Final Issue
30 June 1989

User's Manual

for the

Portable ASP Work Station (PAWS)

Document Number VCW-01843-10-16

Contract No. N62269-86-D-0125

Requisition No. 7170822

Prepared for Code 7011

Naval Air Development Center

Warminster, PA 18974

Prepared by Vitro Corporation

J6K

Table of Contents

Table of Contents (Continued)

Table of Contents (Continued)

Table of Contents (Continued)

Table of Contents (Continued)

Table of Contents (Continued)

Table of Contents (Continued)

List of Figures

Acronym List

AC	Alternating Current
ADCP	ASP Diagnostic Control Program
AECW	Arithmetic Element Control Word
AP	Arithmetic Processor
ASP	Advanced Signal Processor
CMPO	Compare Micro Program Override
CP	Control Processor
CPU	Central Processing Unit
CR	Carriage Return
CS	Control Store
CSAR	Control Store Address Register
CSCI	Computer Software Configuration Items
CW	Control Word
EMI	Electro Magnetic Interface
EST	Eastern Standard Time
EU	Electronics Unit
FDD	Flexible Disk Drive
FPC	Floating Point Coprocessor
FPD	Flat Panel Display
HWCI	Hardware Configuration Item
IC	Instruction Counter
IOM	I/O Monitor Function
LED	Light Emitting Diode
LPI	Laboratory Panel Interface

Acronym List (continued)

MAR	Micro Store Address Register
MC	Machine Check
MCKO	Machine Check Override
MMU	Memory Management Unit
MS	Micro Store
MTS	Microprogrammable Test Set
MVME	Motorola VME
NADC	Naval Air Development Center
PAWS	Portable ASP Work Station
PS	Program Store
PSW	Program Status Word
RAM	Random Access Memory
RDD	Rigid Disk Drive
RTUX	Real Time UNIX Executive
SCON	System Controller
SDE	Software Development Environment
SIOA	System Input/Output Adapter
SOC	Stop On Compare
SOE	Stop On Error
SPL	Signal Processing Language
SYCS	Sync Control Store
SYMS	Sync Micro Store
SYPS	Sync Program Store
TI	Test Integration
TLCSC	Top Level Computer Software Components

Acronym List (continued)

TU	Terminal Unit
VAC	Volts AC
VME	Versa Module Europa

1.0 Scope

1.1 Identification

This user's manual provides the procedures for the standalone operation of the Portable ASP Work Station (PAWS) Terminal Unit (TU). The TU, shown in Figure 1-1, is one of two Hardware Configuration Items (HWCI) which make up the PAWS system. The second HWCI is Electronics Unit (EU). Together, these two units will support the execution of two Computer Software Configuration Items (CSCI) as follows:

1. Host Computer CSCI - This CSCI provides the PAWS operator with a Software Development Environment (SDE).

2. PAWS Controller CSCI - This CSCI provides the user with special purpose Top Level Computer Software Components (TLCSC) to load and control the Advanced Signal Processor (ASP). The PAWS Controller CSCI consists of five functions as shown in Figure 1-2. These five are:

   Microprogrammable Test Set (MTS) Function

   System Input/Output (SIO) Adapter Function

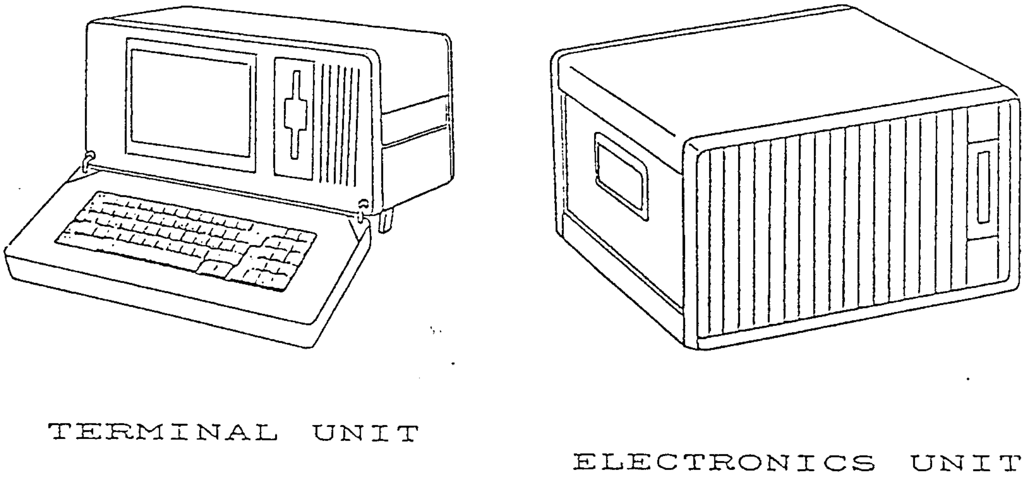
   Test Integration (TI) Function

   ASP Diagnostic Control Program (ADCP) Function

   I/O Monitor Function (IOM)

1.2 Purpose

The purpose of this document is to familiarize the user with the operation of the standalone TU. In the standalone environment, the user will have the SDE and the Microprogrammable Test Set (MTS) function, a sub-function of the PAWS controller CSCI, at their disposal. The other four PAWS Controller Functions require the Electronics Unit (EU) and will not be discussed in this document.



TERMINAL UNIT

ELECTRONICS UNIT

Figure 1-1. Portable ASP Work Station (PAWS)

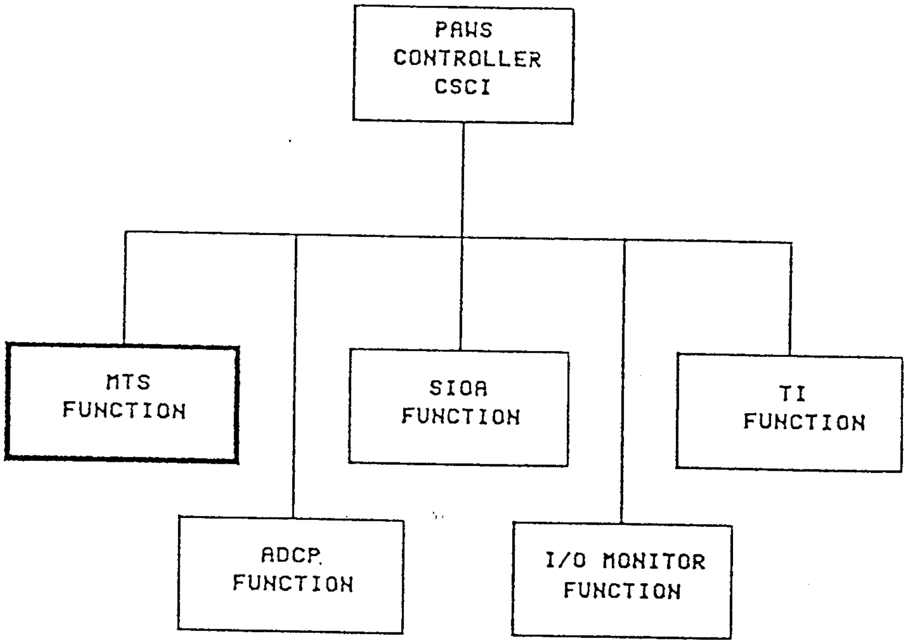


Figure 1-2. PAWS Controller CSCI Sub-functions

1.3 Introduction

This document describes the use of the PAWS equipment and the associated software. This manual begins with a general description of the TU to familiarize the operator with the TU controls and indicators. Next, this manual will describe the recommended startup procedure which will take the user to the PAWS login prompt. From this prompt the user will be shown how to use the Shutdown, the System Administration, the RTUSER1 and the PAWSWORK user spaces. The file utility commands of PAWSWORK and the use of the Microprogrammable Test Set (MTS) software commands will be discussed in detail. The final area to be covered in this document will be preventative maintenance.

Several appendices are provided to help the user as follows:

Appendix A is a quick reference for the PAWSWORK commands.

Appendix B is a quick reference for the MTS commands.

Appendix C is a summary of MTS error messages.

Appendix D is a summary of modem commands.

Appendix E is a summary of onboard diagnostic commands.

Appendix F is a summary of the VI editor commands.

2.0 Referenced Documents

2.1 Commercial Documentation

MVME130DIAG/D2	MVME Diagnostic Firmware User's Manual
MVME130BUG/D4	130Bug Debugging Package User's Manual
MU43815PG/D2	System V/68 Release 3, Programmer's Guide
MU43814PR/D2	System V/68 Release 3, Programmer's Reference Manual
MU43813SAG/D2	System V/68 Release 3, System Administrator's Guide
MU43813SAG/D2	System V/68 Release 3, System Administrator's Reference Manual
N/A	2424STM 2400 Baud Modem Instruction Manual

2.2 PAWS Project Documentation

VCW-01843-10-2	Prime Item B Specification for the Portable ASP Work Station (PAWS) Terminal Unit (TU)
VCW-01843-10-6	Portable ASW Work Station (PAWS) PAWS Controller Computer Software Program Performance Specification (PPS)

3.0 Instructions for Use

3.1 General Description

The Terminal Unit (TU) is a general purpose portable computer. In the standalone mode, a Laboratory Panel Interface (LPI) module will be installed to support the MTS software function. When the TU is received, the TU handle will be in the straight position and the keyboard will be in the closed or folded up position. To release the keyboard for use, the TU handle must be rotated up or down. This can be performed by pressing in on the release buttons on the handle. The handle when rotated down, may be used to support the front of the TU and hold the TU at a 15 degree incline. The two front feet of the TU may be used to serve the same purpose. Once the handle is out of the way, the keyboard may be lowered by pressing in on the two release buttons on the sides of the keyboard enclosure as shown in Figure 3-1. The keyboard may also be removed for further user convenience.

3.1.1 Frontview Description

With the keyboard lowered, the front view of the TU as shown in Figure 3-2 will be visible. The following paragraphs will describe the various components which are visible in this view.

3.1.1.1 TU Keyboard

The keyboard is an 80 position, low profile, full travel membrane keyboard. The keyboard is housed in a fold-down, detachable enclosure with an eight foot, when extended, coiled keyboard cable. The keyboard will serve as the primary input device for the user.

3.1.1.2 Flat Panel Display (FPD)

The FPD is the primary output device for the user interface. The FPD implements electroluminescence technology to provide a 25 row, 80 column display screen. A FPD filter reduces the glare and reflection and also gives the display the amber color. The FPD requires just one third the volume of a conventional CRT display and is much more rugged. Care of this display is described in section 3.4 Preventative Maintenance.

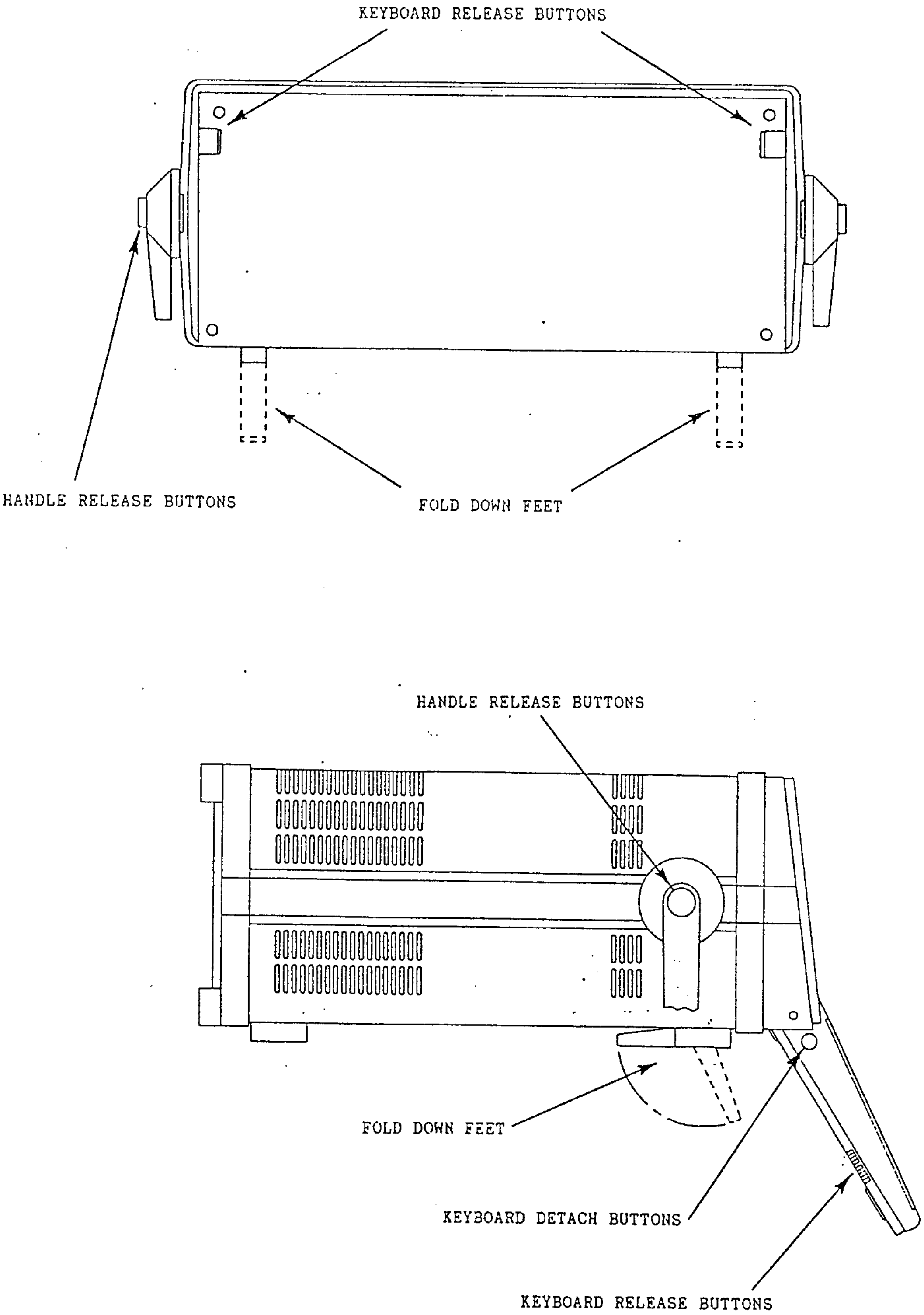


Figure 3-1. TU Setup

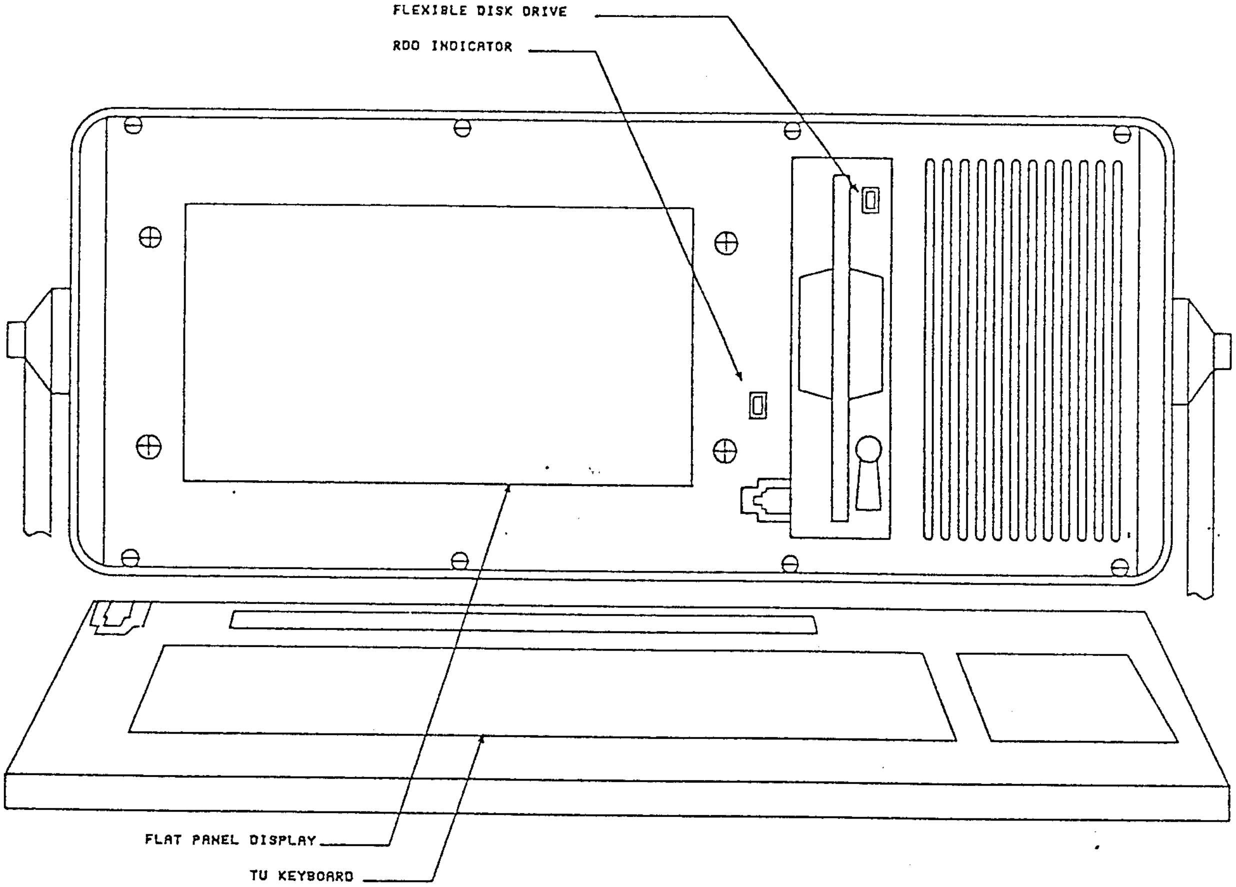


Figure 3-2. TU Enclosure Frontview

3.1.1.3 Flexible Disk Drive (FDD)

The FDD is the primary removable media data storage device available to the user. The FDD provides a two density capability of either 655 Kb or 1.2 Mb densities. In the low density, 655 Kb, the user may use readily available double-sided/double density media; however, for high density operation, 1.2 Mb, the user must use quad density type media. The user will want to use the high density mode for backing up data and files on the Rigid Disk Drive (RDD). The lower density will be used primarily for reading diskettes which have been formatted on a Motorola system running UNIX V.2.6.

To operate the FDD, the user must insert a diskette into the FDD slot with the diskette label side to the right. Once installed, the user must turn the FDD handle in the direction of the arrow shown on the drive itself. A LED indicator on the FDD will show the user when the FDD is reading or writing. The diskette must not be removed when this indicator is on.

3.1.1.4 Rigid Disk Drive (RDD)

The TU has a 42 Mb formatted RDD installed inside the TU which is not visible to the user. A LED indicator has been installed in the TU front panel which will indicate the activity of the RDD to the user. The RDD is the primary offline storage device available to the user. The RDD will contain all the operating system and user files. It is recommended that the user make frequent backups of the user area using the FDD. This should be done on a weekly basis and before the TU is transported anywhere.

3.1.2 Rearview Description

Figure 3-3 shows the rearview of the TU. This view will be further divided into a rearpanel and the TU module description. These will be discussed in the following paragraphs.

3.1.2.1 Rear Panel Description

3.1.2.1.1 AC Power Cord Receptacle

The AC power cord receptacle supports/provides three functions. First, it supports the input of 110 VAC 60 or 400 Hz AC power. Second, it provides Electro Magnetic Interference (EMI) filtering.. The third function is AC fuse protection. Just below the AC receptacle is a cover which provides access to two AC fuses. The fuse type is 3A/250V. These fuses should not be replaced with any other value.

3.1.2.1.2 Power On Switch

The power on switch is located in the upper corner of the TU rear panel. When off, a "0" will be visable to the user on the side of the switch and the switch will not be illuminated. When turned on, the switch will be illuminated, the "0" will no longer be visable and AC power will be applied to the TU power supply.

3.1.2.1.3 Modem Inputs

The TU has a Hayes compatable modem installed internally. The modem is a 2400 baud autoswitching type modem. The modem is configured to the settings shown in Table 3-1. The user may call out to another computer system or the TU may be called by another computer system. Phone line connections are made via the TU rear panel as shown in Figure 3-3. The upper modem jack is reserved for the line connection. The lower jack can be used to plug in a phone.

Table 3-1. Initial Modem Settings

Function	Setting
Auto LF/CR	OFF
Keyboard Echo	OFF
Duplex	FULL
Baud rate	2400
Data Bits Per Character	8
Transmit Parity Enable	OFF
Receive Parity Sense	OFF
Parity Bit	SPACE or 0
Stop Bits	1

3.1.2.1.4 Printer Port

The printer port is located on the rear of the TU as shown in Figure 3-3. It is a serial RS232 type connector. The printer port is a connector located on the rear of the TU. When activated the printer is slaved to the FPD. All information printed on the screen will then be outputted to the printer.

The printer port for the TU operates with the following parameters:

interface type	serial
baud rate	9600
parity	none
data word length	8 bits
stop bits	one

The recommended printer is the Datasouth DS400 or equivalent. The Datasouth Computer Corporation is located at the address listed below:

Datasouth Computer Corporation
4216 Stuart Andrew Boulevard
Charlotte, North Carolina 28210

3.1.2.1.5 Laboratory Panel Interface (LPI) Connector

A military connector for the LPI is recessed in the rear panel. The LPI external cable should be connected to the ASP and TU when the MTS function is in use. If the MTS environment is entered without the cable installed, an error message will be displayed on the screen of the display.

3.1.2.1.6 Air Outlet

The rear panel provides an outlet to exhaust heat from the TU internal electronics.

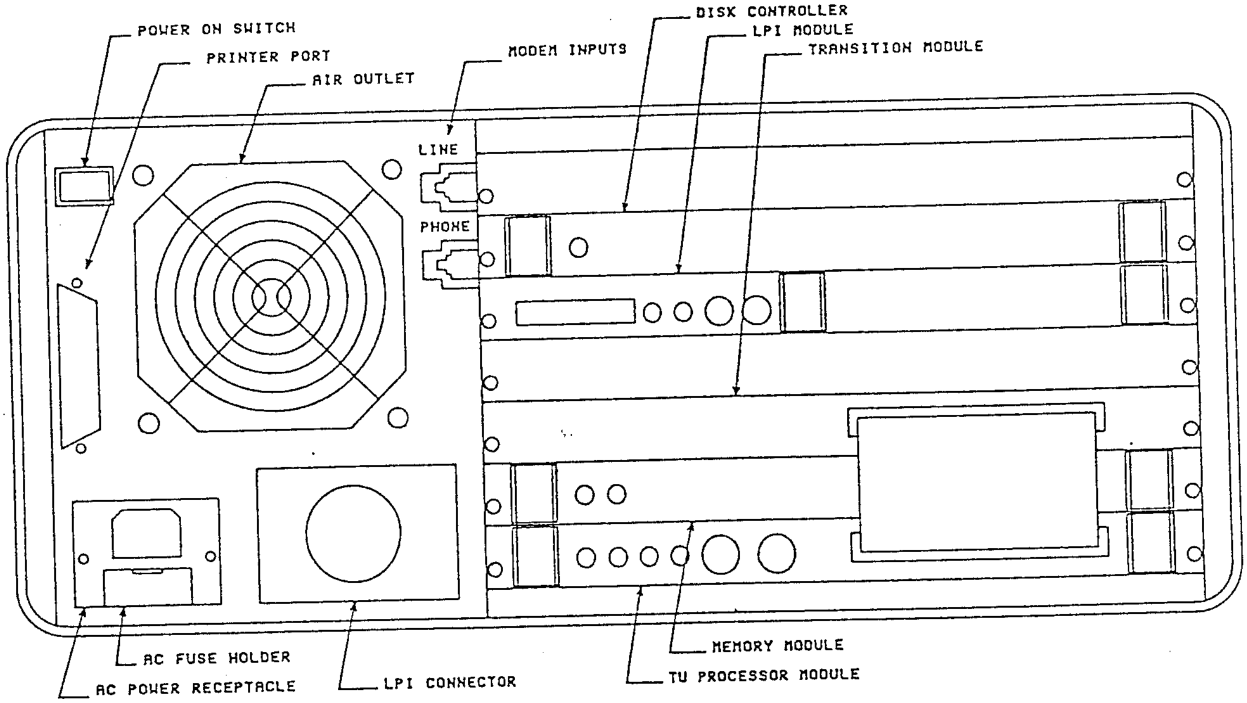


Figure 3-3. TU Enclosure Rearview

3.1.2.2 TU Module Description

The TU modules may be accessed from the rear of the TU as shown in Figure 3-3. The TU will support a total of seven VME bus compatable modules. When delivered, five of the seven available slots will be filled. Slot 4 is reserved for expansion to the Electronics Unit. Slot 7, while appearing unused, is currently configured to the disk drives internal to the unit. The slot numbering system will be in ascending order from bottom to top as shown in Figure 3-4. Several of the modules have indicators and controls which will be discussed in the following paragraphs.

3.1.2.2.1 TU Processor Module

The TU processor is located in the first VME bus slot. This processor is a MC68020 based processor with onboard Floating Point Coprocessor (FPC), Memory Management Unit (MMU), VME130 Debug Monitor (130bug) and cache memory. The processor will support a multiprocessor environment where it is the bus master and provides the primary user interface support for the display, keyboard and disk drives. The controls and indicators are shown in Figure 3-4 and are described as follows:

Abort Switch (S1) - The Abort switch is a momentary type switch that when pressed, causes a Level 7 interrupt to the MC68020 processor. The 130bug treats Abort as a low level reset. The result is a display of the MC68020 registers on the screen and the return of the program control to 130bug.

Reset Switch (S2) - The Reset switch is a momentary type switch that when pushed will cause the local processor and I/O devices as well as any device tied to the VME bus signal "SYSRESET*" to be reset.

System Configuration DIP switch (S3) - Switch S3 is an eight position piano type DIP switch that is accessible through the processor front panel. These switches are preset prior to delivery and should not be changed. The preset values for Switch S3 are: positions 2,6,7,8 are set and the remainder are cleared.

Fail Indicator - The Fail indicator is a discrete red LED that is illuminated by the processor if a processor failure has occurred. When power is initially applied, the Fail indicator will illuminate briefly.

Halt Indicator - The Halt indicator is a discrete red LED that is illuminated when the MC68020/MMU are in a halted state.

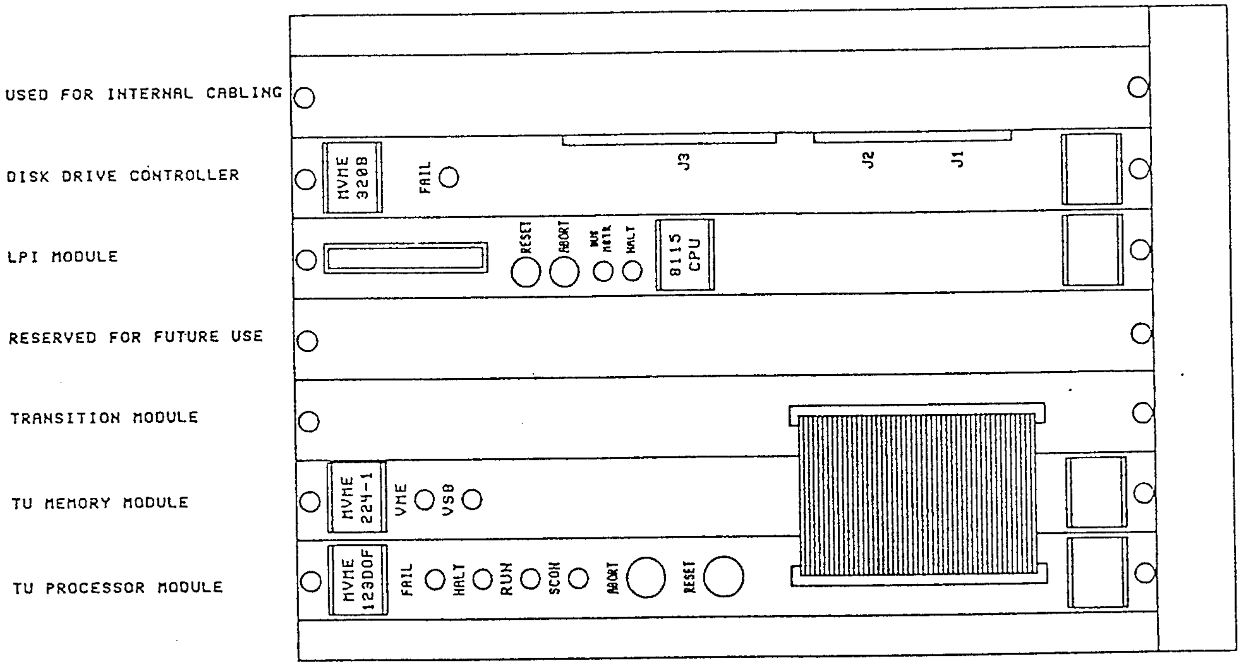


Figure 3-4. TU VME Slot Usage

Run Indicator - The Run indicator is a discrete green LED that is illuminated when the processor is addressing a module on the VME bus.

Scon Indicator - The System Controller (Scon) indicator is a discrete green LED that is illuminated by the TU processor since it is the System Controller.

3.1.2.2.2 Memory Module

The memory module is located in VME bus slot 2 and provides 4 Mb storage capacity. It is a high performance dynamic memory module. The memory module has two indicators as follows:

VME Indicator - The VME indicator is illuminated green when the memory module is being accessed via the VME bus.

VSB Indicator - The VSB indicator is illuminated green when the memory module is being accessed via the VSB bus.

3.1.2.2.3 Transition Module

The transition module is located in VME bus slot 3 and converts the TTL RS232 interfaces from the TU processor to compatable interfaces for the TU display and internal modem. These signals are passed through a ribbon cable that can be seen from the rear of the unit. There are no indicators or controls associated with this module.

3.1.2.2.4 Disk Controller Module

The disk controller module is located in VME bus slot 6 and provides the interface support for the RDD and the FDD. There is one indicator on the controller front panel as follows:

Fail Indicator - The Fail indicator is a discrete red LED that is illuminated by the controller if a controller failure has occurred.

3.1.2.2.5 LPI Module

The LPI module is installed in the VME bus slot 5. This module provides the LPI interface used to communicate with the ASP to support maintenance activity functions. The front panel contains the following controls and indicators.

Abort Switch (S1) - The Abort switch stops processing on the LPI module.

Reset Switch (S2) - The Reset switch is a momentary type switch that when pushed will cause the local processor to be reset. This is the equivalent of a warm start.

Halt Indicator - The Halt indicator is a discrete red LED that is illuminated by the processor if a processor failure has occurred. When power is initially applied, the Halt indicator will illuminate briefly.

BusMSTR Indicator - The Bus Master indicator is a discrete red LED that is illuminated when the processor is addressing a module on the VME bus.

Serial Port - A 26 pin serial connector is provided on the front of the module.

3.1.3 Cabling Requirements for the TU

The TU requires very little cabling when in the standalone mode. The user will be required to install the AC power cord, the LPI cable, modem connections, and the printer cable if needed.

3.1.3.1 Installing the AC Power Cord

The TU uses an industry standard AC power receptacle and is delivered with a molded three wire AC power cord. This cable has a standard three prong plug to be plugged into a 110 VAC wall outlet. If the user takes the TU onboard an aircraft, the TU can operate on 400Hz power; however, the user may have to change the three prong plug to an aircraft compatible plug.

The TU supports the AC power requirements shown in Table 3-2.

Table 3-2. AC Power Requirements

Volts	Frequency
115 VAC	60 or 400 Hz
(90 to 132 VAC)	(47 to 440 Hz)

3.1.3.2 Installing the LPI Cable

In order to run the Microprogrammable Test Set (MTS) software the TU must be cabled to the ASP via the LPI port. The TU is delivered with a LPI cable for this purpose. The cable is labeled as to which end is installed to which equipment. This cabling requirement is shown in Figure 3-5.

3.1.3.3 Installing the Modem Connections

The modem is connected via standard 6 pin modular jacks. The phone line from the wall should be connected to the modem connector labeled "LINE". A standard phone may be connected to the modem connector labeled "PHONE".

3.1.3.4 Installing the Printer Cable

A printer cable may be installed from the TU printer port located on the rear panel to a serial printer.The location of the TU printer port is shown in Figure 3-3.

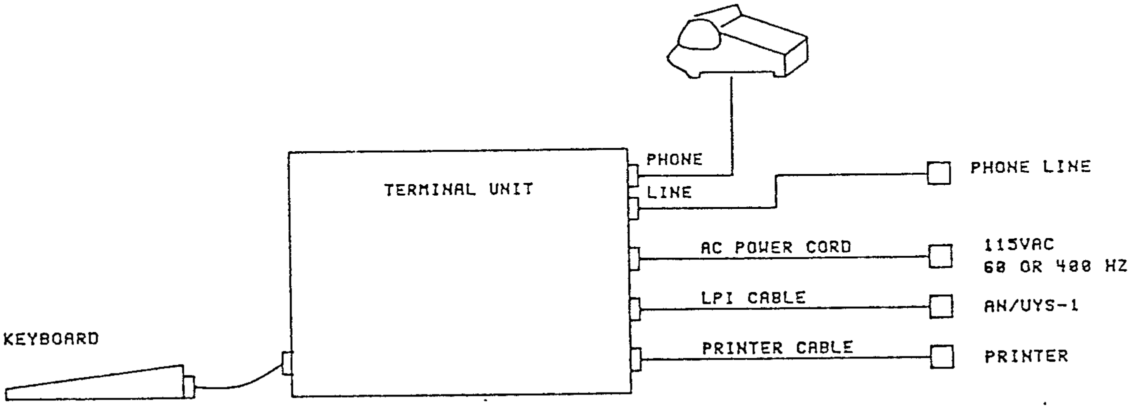


Figure 3-5. TU Cabling Requirements

3.2 TU Operation

The TU startup begins with the application of AC power. This is accomplished by switching the power switch to the "ON" position.

3.2.1 TU Startup Procedure

When power is applied, the TU processor module onboard ROMS are enabled and the following information should be displayed on the FPD.

```
FPC passed test
PMMU passed test
130Bug>
```

The 130BUG prompt is the entry point for the onboard debug monitor. The 130BUG supports both a debug and diagnostics mode. The mode can be switched by typing "sd <CR>". The debug monitor is used for the debug of application software being executed in the TU processor and will not be discussed in this document since the software delivered with the TU has already been developed. The 130DIAG will be used to diagnose problems with the TU off-the-shelf modules. These diagnostics are discussed in Appendix E.

3.2.2 Booting UNIX

The UNIX may be booted from the "130Bug>" prompt. The RDD must be up to speed prior to booting UNIX. This will take approximately 15 seconds. When the RDD is up to speed, the head restraint solenoid in the RDD is activated and a "click" can be heard. The RDD is now up to speed.

The user may type "bo<CR>" at the "130Bug>" prompt.

If the user types "bo" prior to the RDD spinning up the following error message is displayed

```
Boot Error
Packet Status 0002
130Bug>
```

In response to this message, the user should wait a few more seconds and reenter "bo".

The boot process will begin with

```
IPL loaded at: $000F0000

***************************************************

System V/68 Release R3V4 68020 Version 871216

     Real Mem=4194304
     Avail Mem=2920448

**************************************************

Is the date Mon Jan 9 13:34:02 EST 1989 correct?(y or n)
```

If any key, except "y", entered, the user will be prompted to enter the correct date and time as follows:

Enter the date as MM/DD/YY:hh:mm:ss

The user does not have to enter the "/" or ":", any non-numeric delimiter will do.

At this point UNIX and other system files will be loaded. The last process to be loaded is the Real Time UNIX Executive (RTUX) used when operating ASP specific software.

The following prompt will now appear

```
*** Wait for RTUX confirmation before logging in ***
```

PAWS login:

After a brief delay the display will yield:

PAWS Login: RTUX Executive loaded V3.2+.V
Copyright (c) 1985 by Emerge Systems, Inc.

At this point a <CR> entry will display the PAWS Login as follows:

PAWS login:

The booting process is now complete. The UNIX environment has been divided into four user areas as follows: the Unit Shutdown, the System Administrator, the RTUSER1 and the PAWSWORK areas. The capabilities and use of these user areas will be discussed in the following paragraphs.

3.2.3 Using the Shutdown Procedure

The shutdown procedure must be implemented whenever the TU is to be powered down. This will ensure that the file systems are correctly preserved. From the PAWS login prompt enter:

```
PAWS login:  shutdown
```

This login will invoke the shutdown procedure and the following information will then be displayed.

```
     SYSTEM V/68 Release R3V4  Version 871216  M68020
     PAWSv68
     Copyright (c) 1984 AT&T
     Copyright (c) 1985 Motorola, Inc.
     All Rights Reserved



     SHUTDOWN PROGRAM

     Thu Apr 13 09:45:24 EDT 1989
     Broadcast Message from shutdown (console) on PAWSv68 Thu
Apr 13 09:45:24...

     System coming down in 10 seconds ! ! !

     PLEASE LOG OFF NOW ! ! !

     Broadcase Message from shutdown (console) on PAWSv68 Thu
Apr 13 09:45:34...

     SYSTEM BEING BROUGHT DOWN NOW ! ! !

     You may power down the system now - it's been fun
computing with you!
```

3.2.4 The System Administrator

A System Administrator should be assigned to the TU. One of the system administrator's tasks is to assign UNIX working spaces for TU users. Once assigned the user's may assign a password to the working space; however, it is imperative that these passwords are kept in a secure place. This will prevent a situation in which the passwords are misplaced or forgotten resulting in an inability to access the system.

To support the System Administration function, the TU is delivered with a very useful and user friendly software package called the System Administrator Software. It may be used to perform various system administrator functions. Most of these functions have been configured for the TU and will not require reconfiguring. If the user would like to become more familiar with the System Administrator Software, the user should read the System Administrator's Guide listed in the reference documentation.

3.2.4.1 Invoking the System Administrator Software

To invoke the System Administrator Software, the user must login as follows:

```
PAWS login: sysadm
```

After logging in, the System Adminstrator's menu will be displayed as shown in Figure 3-6. The user may select the desired system administration task to be performed and the system administrator will be stepped through the selected function until the task has been completed. The system administrator for the TU will primarily be concerned with the Disk Management and File Management functions.

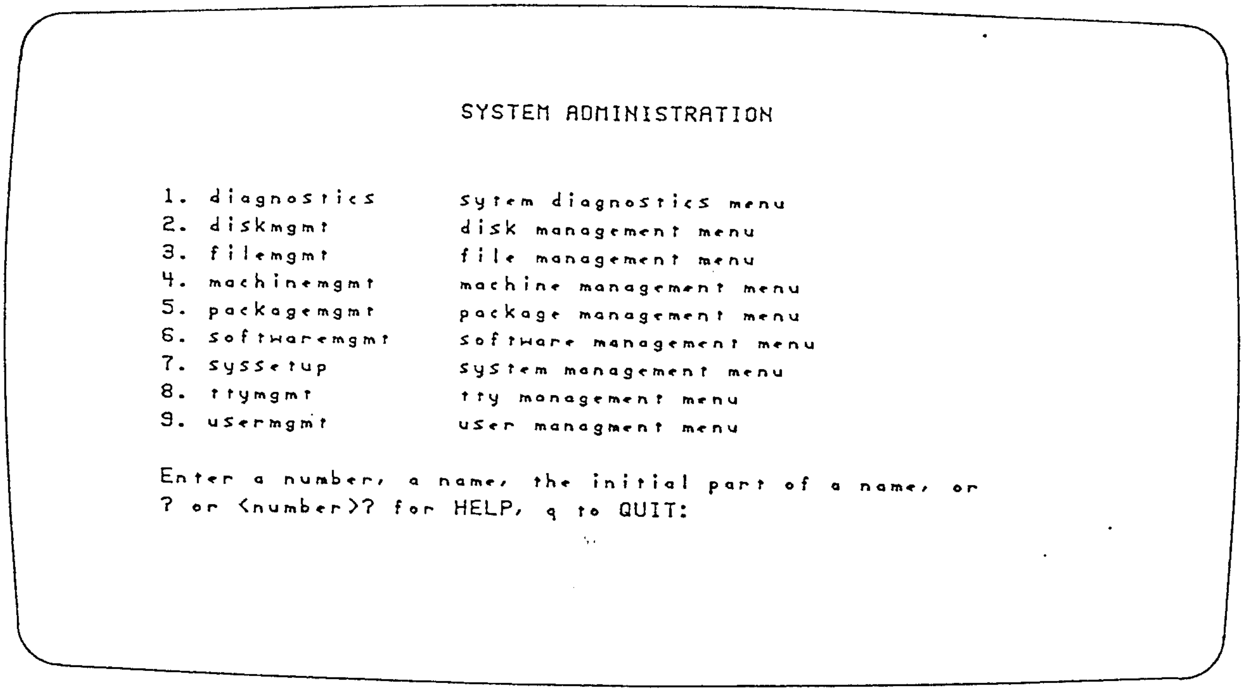
3.2.4.2 Major functions of the System Administrator

Diagnostics

* 1) gives advice on repair of built-in disk errors
* 2) report on built-in disk errors

Disk Management

1) check a removable medium file system for errors
2) make exact copies of a removable medium
* 3) erase data from removable media
4) format new removable diskettes
* 5) hard disk setup
6) create a new file system on a removable medium
7) mount a removable medium file system
8) unmount a removable medium file system



```
                    SYSTEM ADMINISTRATION

1. diagnostics      sytem diagnostics menu
2. diskmgmt         disk management menu
3. filemgmt         file management menu
4. machinemgmt      machine management menu
5. packagemgmt      package management menu
6. softwaremgmt     software management menu
7. syssetup         system management menu
8. ttymgmt          tty management menu
9. usermgmt         user managment menu

Enter a number, a name, the initial part of a name, or
? or <number>? for HELP, q to QUIT:
```

Figure 3-6. System Administrator's Menu

File Management

1) backup files from built-in disk to removable media; read files from hard disk
2) display how much of the hard disk is being used
* 3) list file older than a particular date
* 4) list the largest files in a particular directory

Machine Management

* 1) stop all running programs then enter firmware mode
* 2) stop all running programs then turn off machine
* 3) stop all running programs then reboot the machine
4) print list of users currently logged onto the system

Package Management

* 1) basic networking utilities menu

Software Management

1) install new software package onto built-in disk
2) list packages already installed
3) remove previously installed package from built-in disk

Note: capability to be utilized at future time

System Setup

1) assign or change administrative passwords
2) set the date, time, time zone and daylight savings time
3) set the node name of this machine
4) set up your machine the very first time
5) assign system passwords

Note: used in assigning passwords

TTY Management

* 1) show tty line settings and hunt sequences
* 2) create new tty line settings and hunt sequences
* 3) show and optionally modify characteristics of tty lines

User Management

1) add a group to the system
2) add a user to the system
3) delete a group from the system
4) delete a user from the system
5) list groups in the system
6) list users in the system
7) modify defaults used by adduser
8) menu of commands to modify group attributes
9) menu of commands to modify a user's login

* Not used in normal operation of TU

3.2.4.3 Using the Disk Management Function

The system administrator will be required to initialize diskettes for use with the TU. This includes diskette preparation for backing up and archiving purposes and diskette preparation for normal TU operation when working in the UNIX environment or when operating PAWSWORK. The system administrator may gain access to the diskette initializing functions through the Disk Management Menu.

When preparing diskettes it is important that the system administrator understand the following terms:

Format - Create tracks and sectors

Make file system - To write identifying labels on the magnetic medium so the system can know what is brought on line. Also, to define file systems that are removable for reasons of privacy or security.

Mount/unmount - To bring a file system under operating system control (mount) or to release (unmount).

When Archiving files, the system administrator will use the UNIX command called "cpio". For this command, directories are not required to be put on the diskette. Therefore, the system administrator only needs to format the diskettes and does not have to make a file system nor mount the FDD. Software updates for the TU will be delivered to the system administrator on diskettes using the "cpio" command.

When diskettes are needed for PAWSWORK or when using the UNIX copy (i.e., cp) command, a directory must be created on the diskette once the diskette has been formatted. This will require that all three diskette initialization steps be performed.

1) formatted
2) filesystem made
3) unmounted if necessary

When entering PAWSWORK, the user will be asked if a floppy diskette will be used during the PAWSWORK session. If the user does not have a diskette properly prepared by the system administrator as listed above, the user may prepare a blank diskette in the PAWSWORK by answering "yes" to the appropriate questions.

3.2.4.4 Using the File Management Function

The system administator will use the File Management function to manage the use of the Rigid Disk Drive (RDD). As users work with the TU, such as creating print files in PAWSWORK, the disk space will decrease. The system administrator can view the disk usage from the File Management function and determine the largest files on the RDD. The system administrator may then take the appropriate action to regain disk space. This is normally done by archiving older files and deleting them from the RDD.

3.2.4.5 Exiting the System Administrator Function

To exit the system administrator section enter 'q'. Similiarly, when in a subsection of the system administrator menu a 'q' or a <CR> is entered to return to the main menu where the exit command can then be utilized.

3.2.5 Logging in as a UNIX User

For users familiar with the UNIX operating system, the TU is configured with a pre-established user space called RTUSER1. This space may be accessed by logging in as follows:

```
PAWS login: rtuser1
```

This working space does not have a password. Though one could be assigned, it is recommended that this space be set aside for general user use. If a user requires a space with password protection, the system administrator may set up user specific working space and the user may then assign a unique password.

Once logged in, the user will have UNIX system V.3 at their disposal. The TU is delivered with a "C" compiler, assembler for the 68010 and 68020, linkers and loaders. The user may access the FDD, the Printer Port, and the Modem with standard UNIX commands. The UNIX commands associated with these peripherals will be discussed in this section. The use of other UNIX commands may be accessed from UNIX referenced documents or by using the UNIX online help fucntion. Online help may be accessed by entering "**help**"at the UNIX prompt.

3.2.5.1 Using the Flexible Disk Drive (FDD) in UNIX

The UNIX user may use the FDD for backing up and restoring files from the RDD. This activity consists of two steps: preparing FDD diskettes and copying files. Both will be discussed in the following paragraphs.

3.2.5.1.1 Preparing FDD Diskettes

As UNIX user, diskettes may be prepared for archiving (i.e., when using the UNIX CPIO command) or with directories (i.e., when using the UNIX CP command or when using diskettes for PAWSWORK). The following paragraphs summarize the UNIX diskette preparation commands:

Initialize/Format a Diskette

```
$/etc/fmtflp -v /dev/rSA/hddiskette1
```

Make a Filesystem on the Diskette

```
$fs /dev/SA/hddiskette1
```

Label a Diskette

```
$/etc/labelit /dev/SA/hddiskette1 fd paws
```

Once initialized, the user must mount the diskette. This will inform UNIX that the FDD is available for use.

Mount a Diskette

```
$mnt hdd
```

Note: When the user is finished with the session, the user must unmount the diskette or files collected during that session may be lost.

Unmount a Diskette

```
$umnt hdd
```

When working in PAWSWORK, the diskette will be mounted automatically when the user answers "yes" to the "Do you wish to use the Floppy?" prompt when entering PAWSWORK. When existing PAWSWORK, the diskettes will be automatically unmounted.

3.2.5.1.2 Copying Flexible Disk Drive (FDD) Files in UNIX

The FDD is the off-line storage device for the TU. Once the user has prepared the diskettes, the user may backup files on the RDD to the FDD or restore files from a FDD diskette back to the RDD. The files may be stored to the diskette using either the UNIX archive command, cpio, or using the UNIX copy command, cp.

The UNIX archive command, cpio, optimizes the amount of information that may be stored on the diskette. To use the CPIO command the diskette only needs to formatted and does not need a file system. There will be no directory available once the files have been archived.

The following cpio command may be used to archive the file 'test1' from the RDD to the FDD.

```
$ls test1 | cpio -ocv > /dev/rSA/hddiskette1
```

The cpio command is an archive function which does not create a directory when executed. The file 'test' must be in the current directory. For example, if the file 'test' is a print file created during a PAWSWORK session, the directory must be changed to pawswork since all files created during pawswork are stored in the pawswork subdirectory. The following command may be used:

```
$cd /usr/pawswork
```

The following cpio command will restore all information archived on the FDD back onto the RDD.

```
$cpio -icvd < /dev/rSA/hddiskette1
```

The UNIX copy command may be used to copy files from the RDD to the FDD and copy them back. When using this command, a directory will be available to the user which means that the diskette must have been prepared with a file system and mounted prior to use. The following command will copy a file 'test1' from the RDD to the FDD:

```
$cp test1 /fd
```

The following command will copy a file 'test1' from RDD to FDD and rename the file 'test2':

```
$cp test1 /fd/test2
```

The following command will copy all files from one subdirectory on the RDD to the FDD:

```
$cp * /fd
```

The following command may be used to copy a file 'test1' from the FDD to the present directory on the RDD.

```
$cp /fd/test1 test1
```

The following command may be used to copy all files on the FDD to the /usr/pawswork subdirectory on the RDD.

```
$cp /fd /* /usr/pawswork
```

The following command may be used to copy all files on the FDD to the present directory on the RDD.

```
$cp /fd* *
```

3.2.5.2 Using the Printer Port

Prior to turning on the power the printer must first be properly cabled to the printer port as described earlier.

3.2.5.2.1 Using Print_On and Print_Off Commands

The TU is delivered with two print commands "print_on" and "print_off" located in the RTUSER1 user space. Both of these commands are special purpose commands developed to enable/disable the printer port on the Flat Panel Display (FPD). When enabled, all information displayed on the FPD will also be sent to the printer via the RS232 port on the rear panel of the TU. To activate the printer port the user must be in the RTUSER1 subdirectory. At the UNIX prompt, the user should type:

```
$ print_on
```

Note: Do not enable this port unless a printer has been cabled to the printer port, otherwise, the system may hang.

To disable the printer port the following command must be executed.

```
$ print_off
```

To print a file created during a PAWSWORK session, the following steps should be taken:

1. Enter the RTUSER1 user space via login or by changing directory if already logged into a different directory. Enter print_on at the UNIX prompt.

```
$ print_on
```

2. Change the directory to the PAWSWORK directory where all files created during a PAWSWORK session are stored by entering the following command at the UNIX prompt.

```
$ cd /usr/pawswork
```

3. Cat the file to be printed to the display. This may be performed by entering the following command:

```
$ cat test1
```

The printer should begin to print as the first characters are displayed.

When printing has been completed, the printer port should be disabled. To disable the printer port the directory must be changed back to the RTUSER1 working space and the print_off command executed.

```
$ cd /usr/rtuser1

$ print_off
```

Printing should now be halted.

3.2.5.3 Using the Internal Modem

To use the modem, the user must login as a UNIX user (i.e., rtuser1). The UNIX CU command is used to establish communication with the modem as follows:

```
$ cu -l/dev/tty01<CR>
```

"CONNECTED" will be displayed which means the user is now communicating directly with the modem. Once connected, the user may use the Hayes command set to continue to talk to the modem. The Hayes command protocol requires that an "AT" to precede each command. By typing simply "AT<CR>", the modem returns the response "OK" to the user. If an "OK" does not appear, the modem echo may be disabled. To enable echo, the user should enter:

```
ATE1Q0&W<CR>
```

E1 - enables echo

Q0 - disables quiet

&W - saves these new values permanently in the modem's memory.

3.2.5.3.1 Reconfiguring the Modem

The user does not need to change the configuration, unless the user is trying to communicate with a modem that has been configured differently than that defined in initial DTE settings Table. The modem may be reconfigured using this same approach described above; however, after entering "AT", the user should enter the new configuration commands. The configuration commands are summarized in Appendix D. If the new configuration is to be the final configuration, the user must end the command string with "&W".

3.2.5.3.2 Calling Out to Another Modem

To call another modem, the user should enter the following command:

```
ATDT(phone number)<CR>
```

DT is the Dial Tone command for DTMF tone generator.

Phone Number is the number to be called. Note: If the phone line is not dedicated, a "9," may have to precede the phone number

When connection to the receiving modem has been made, the following prompt will be displayed:

```
CONNECTED (baud rate)
```

where the baud rate is the final transmission rate. The TU modem is a 2400 baud modem; however, if the receiving modem has a slower baud rate, the TU modem will switch-down until a compatable baud rate has been established. It is this agreed upon baud rate that will be displayed after the CONNECTED prompt.

3.2.5.3.3 Disconnecting

The user should log off the external system that the user is connected to. This usually causes the modem to disconnect. A "Disconnected" prompt will be displayed. The user will still be connected with the modem. To return to the UNIX prompt enter "~." ("~" represents the tilde key on the keyboard and "." represents a period). The following information will be displayed prior to returning to the UNIX prompt.

```
~[PAWSv68].

Disconnected
$
```

3.2.5.4 Exiting the UNIX Function

To leave the UNIX user space and return to the PAWS login prompt enter :

```
$^D or "bye"
```

3.2.6 Logging in as a PAWSWORK User

To run the PAWS Controller Software the user may login as a PAWSWORK User. This is accomplished by typing "pawswork" at the PAWS login prompt as follows:

```
PAWS login: pawswork
```

Upon entry the user will be asked if a floppy diskette will be used. This will be followed by the entry into the PAWSWORK Operator Interface characterized by the PAWSWORK menu shown in Figure 3-7. The use of the floppy diskette and the PAWSWORK Operator Interface will be discussed in the following sections.

3.2.6.1 Preparing a Floppy Diskette for PAWSWORK

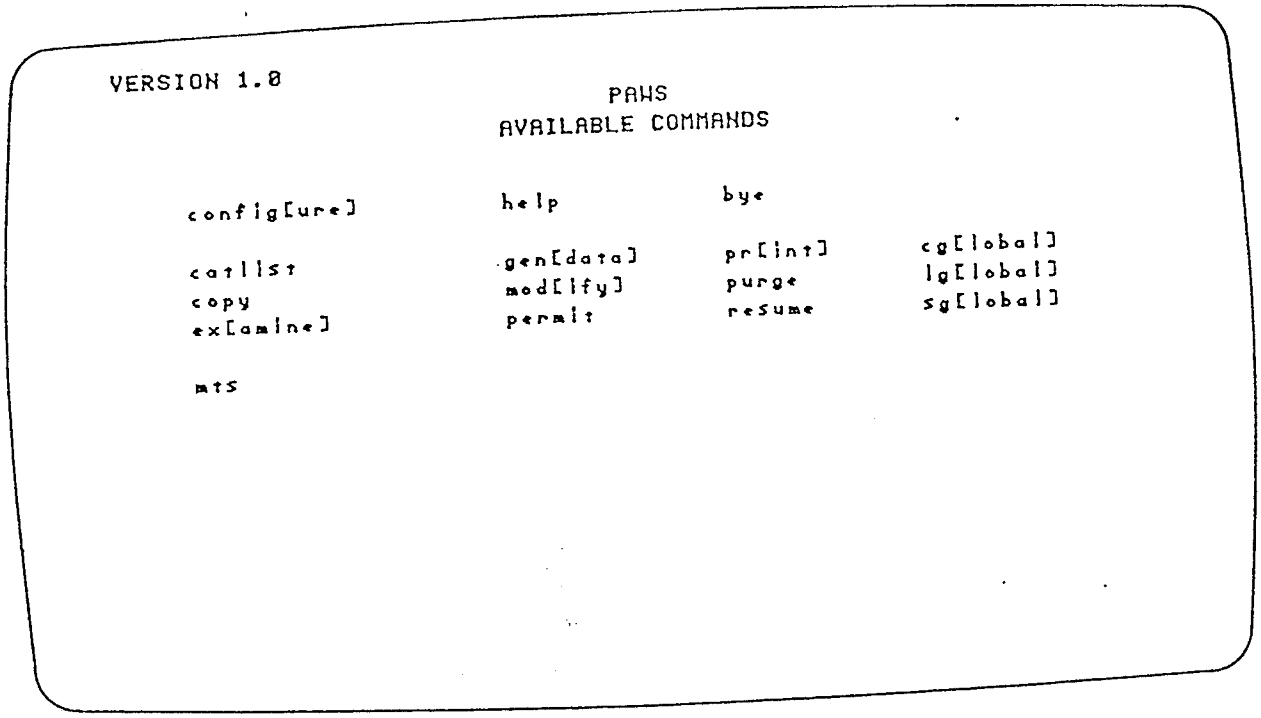
When a user answers "no" to the question for using the floppy, the steps for preparing a diskette will be bypassed and the user will be placed directly into the PAWSWORK Operator Interface software.

When a user answers "yes" that a floppy will be used, the user will stepped through a series of prompts which will define the type of media, to format the media and to make a file system. This sequence ends by mounting the floppy for use during this PAWSWORK session.

Selecting the Right Density - The user must select the correct density for the diskette being used. The user will have two choices: normal or high density. Normal density must be selected when using double-sided, double density type diskettes. This media will provide 655 Kbytes of capacity. High density must be selected when using quad density diskettes. This media will provide 1.2 Mbytes of storage capacity. If the wrong density is selected, an error will be detected and the user will be returned to the PAWS login.

Formatting the Diskette - After the density has been selected, the user will be asked, "Do you wish to format the floppy (y/n)?". If the diskette inserted in the disk drive is blank or contains obsolete files, the user should respond with "Y". If a "Y" is entered the format process would begin. The user may have obtained a diskette from the system administrator which has already been formatted. In this case the user may answer "N". When "N" is entered the software will continue onto the next step.

Creating a File System - A file system must be created on the diskette for use with PAWSWORK. This will automatically be performed for the user. The disk will now be initialized.



```
VERSION 1.0
                                PAWS
                         AVAILABLE COMMANDS

      config[ure]           help          bye

      catlist               gen[data]     pr[int]       cg[lobal]
      copy                  mod[ify]      purge         lg[lobal]
      ex[amine]             permit        resume        sg[lobal]

      mts
```

Figure 3-7. PAWS Operator Interface Menu

Mounting the Floppy - With a properly initialized diskette, the floppy must be mounted for use. This again will automatically be performed for the user during this sequence of steps. Once mounted, the diskette must not be removed until the floppy is unmounted.

Unmounting the Floppy - The PAWSWORK user does not have to worry about unmounting the floppy. The unmount will automatically be performed when the PAWSWORK user leaves PAWSWORK.

3.2.6.2 Using the PAWSWORK Operator Interface

Once the PAWSWORK user has set up a FDD diskette, the interface software for the Laboratory panel Interface (LPI) module will be downloaded and PAWSWORK will be entered.

The PAWSWORK user will be able to select any of the sixteen commands listed in the PAWSWORK Operator Interface menu shown in Figure 3-7. The PAWSWORK commands are organized into four command categories as follows:

1. File Utility Commands

2. PAWSWORK Control Commands

3. Global Commands

4. PAWS Controller Software

The file utiliy commands comprise the majority of commands available to the user. These Operator Interface commands are sumarized in Appendix A.

3.2.6.2.1 File Utility Commands

To invoke any file utility command, the user must enter a proper command syntax at the **PAWS?** prompt. The proper command syntax is dependent on the command to be executed; however, the user must be familiar with some general rules for using the file utilities. Optional parameters are defined by square brackets[].

Specifying a PATH

Files are organized in tree-structured directories. Directories are themselves files that contain information on how to find other files. A PATH to a file is a text string that identifies a file by specifying a path through the directory structure to the file. Syntactically it consists of individual file name elements separated by the slash character.

For example, in /usr/pawswork/test the first slash indicates the root directory tree, called the root directory. The next element, **usr/**, is a subdirectory of the root, **pawswork/** is a subdirectory of usr, and **test** is a file or subdirectory in the directory pawswork.

In this example, **/fd/test**, **fd/** has been set up in the file utilities as a device path to the Flexible Disk Drive (FDD). Thus **test** is a file located in the FDD directory.

Defining a File Name and File Access

The NAME for a file is an arbitrary character string composed of upper and lower case alphabetics, numbers, the underscore (_) and the period (.) characters.

When any PAWS function accepts a user specified filename designated to be written upon, the file utilities function will determine if that filename already exists. If it does not exist, the file will automatically be created with read permission to the user. If the file does exist, the file utilites function will determine if the user has write permission of that file. If not, the file utilities function will display "NO PERMISSION TO WRITE ON EXISTING FILE". If the user does have write permission, the file utilites function will prompt "DUPLICATE FILE NAME XXXXXX" and query the user "REWRITE?". A response of N[O] will invoke the aforementioned error prompt while a response of Y[ES] will permit the user to continue. Access to a file may be changed with the Permit command.

When any PAWS function accepts a user specified filename designated to be read from, the filename must exist and be accessible to the user or "FILE XXXXXX NOT FOUND" message will be displayed.

Specifying a Format Character

The FORMAT Character is a user defined parameter which specifies to the file utilities software the conversion to be performed during the execution of the file utility commands. For the COPY and PRINT commands, the format conversions will be performed on the source data before it is written to the destination. For the EXAMINE command, the format conversions will be performed on the source data prior to examination. For the GENDATA command, the format conversions is applied to the user input before it is written to the destination. The legal parameters for the Format Character are as follows:

a. U - No reformatting performed - Default.
b. C - Compressed Text (Blank text lines are deleted)
c. H - Hexidecimal Dump of binary Data

Creating and Editing Files

The user may create and edit files with the file utility commands. In order to perform these tasks, the VI Editor is used to make any changes or additions in the file chosen. These editor commands are located in Appendix F of this document.

3.2.6.2.1.1 CATLIST

Command Description

The CATLIST command will allow a user to display a file, a directory name or selected files using wildcards as specified by the source. If a directory name is specified, a listing containing the names of all files in that directory will be produced on the display. If a file name is specified a listing will be produced on the display containing detailed data on all files in the specified directory. A wild card character (*) may be used in the path to select common files for display. If a source is not specified, the default will be the pawswork directory on the RDD: /usr/pawswork. If a Place is not specified, the default will be the TU display.

Command Syntax

```
catlist [s[ou]rce=path1] [place=path2]
```

where the parameters are defined as follows:

```
path1 - the source filename or directory
          (default = /usr/pawswork directory on the RDD)

path2 - the destination filename or device name
          (default = "/dev/tty, i.e. TU display")
```

Onscreen Help Information

```
PAWS? help cmd=catlist
```

Example 1

In this example, catlist by itself is entered which will display all files in the pawswork subdirectory.

```
PAWS? catlist

demo1.lpi
demo2.lpi
test1.lpi
test2.lpi
test3.lpi
test1.pdc
```

Example 2

In this example the wildcard character (*) is used to select only the lpi test files from the pawswork directory.

```
PAWS? catlist srce=test*.lpi

     -rw-rw-rw 1 pawswork real_time 50 Jun22 17:08 test1.lpi
     -rw-rw-rw 1 pawswork real_time 50 Jun22 17:08 test2.lpi
     -rw-rw-rw 1 pawswork real_time 50 Jun22 17:08 test3.lpi
```

Example 3

In this example all files on the floppy diskette will be displayed.

```
PAWS? catlist srce=/fd

lpitest.10
lpitest.11
lpitest.13
lpitest.14
```

3.2.6.2.1.2 COPY

Command Description

The COPY command will allow a user to copy files from a source file or device to a destination file or device. The copy operation may be interrupted by pressing the DEL key on the keyboard. The user may also specify a format character which will reformat the data being copied before it is written to the destination. The default source is the pawswork directory on the RDD, the default destination is the pawswork subdirectory of the RDD and the default format character is for no reformatting.

Command Syntax

```
copy s[ou]rce=path1 dest[ination]=path2 [form[at]=fchar]
```

where the parameters are defined as follows:

path1 - the source filename or device to be copied.
(default = /usr/pawswork)

path2 - the destination filename or device to be copied
to (default = /usr/pawswork)

fchar - reformating specification character
(default = u)

Onscreen Help Information

```
PAWS? help cmd=copy
```

Example

In this example the file named test1.lpi will be copied from the RDD to the FDD.

```
PAWS? copy srce=test1.lpi dest=/fd/test1.lpi
```

3.2.6.2.1.3 EXAMINE

Command Description

The EXAMINE command allows a user to examine a file. When displayed, the user will be able to edit the file using the VI editor as summarized in Appendix E. These changes however, will not be saved when the examine is terminated. If an edited file is necessary, the user should use the modify command.

To use this command, the user must have read permission for the file being examined. In addition, the user may specify a format character which will reformat the file prior to the information being displayed.

Command Syntax

```
ex[amine] file=path1 [form[at]=fchar]
```

where the parameters are defined as follows:

path1 - the source filename

fchar - the reformat specification character

Onscreen Help Information

```
PAWS? help cmd=ex
```

Example

This example will display the file named test.10 on the screen of the TU display. To page through this file the user may press ^U or ^D keys.

```
PAWS? ex file=test.10
```

3.2.6.2.1.4 GENDATA

Command Description

The GENDATA command will allow the user to create or generate a data file. This may be done by specifying a destination where to place the file once created. The default is the pawswork directory of the RDD. A format character may also be specified which will reformat the gernerated file prior to writing the file to the destination. This file may be used to create a batch file for execution when operating in the MTS function. If the destination file specified already exists, an error message is given.

Command Syntax

```
gen[data] dest[ination]=path2 [form[at]=fchar]
```

where the parameters are defined as follows:

path2 - the destination filename or directory

fchar - the format parameter

Onscreen Help Information

```
PAWS? help cmd=gen
```

Example

In this example the file test.lpi is created to execute batch commands when in MTS.

```
PAWS? gen dest=test.lpi
```

Upon entry, the user may use the VI editor to generate this file. These VI editor commands are summarized in Appendix E. When complete, the user must enter ZZ to save text.

3.2.6.2.1.5 MODIFY

Command Description

The MODIFY command will allow the user to modify an existing file as specified by file. The user may also specify the destination if the destination will be different than that specified by file. The user must have write privelege to that file or an error message will be displayed. The default for the source is the pawswork directory on the RDD:/usr/pawswork.

Command Syntax

```
mod[ify] file=path1 [dest[ination]=path2]
```

where the parameters are defined as follows:

path1 - the source filename

path2 - the destination filename

Onscreen Help Information

```
PAWS? help cmd=mod
```

Example

In this example, the file named test1.lpi will be modified. When editing is complete, the file will be renamed lpitest.1 and stored onto a floppy diskette.

```
PAWS? mod file=test1.lpi dest=/fd/lpitest.1
```

3.2.6.2.1.6 PERMIT

Command Description

When a file in pawswork is created, the file will be assigned an initial permission of "read" to that user and inaccesssable to others. The originating user may use the PERMIT command to alter these permissions by specifying an access code. The rw/rw permision must be set to transfer files over the modem line.

Command Syntax

```
permit file=path1 access=access code
```

where the parameters are defined as follows:

path1 - the source filename or device for which file permissions are being changed.

access code is one of the following:

rw	Read-Write to user, inaccessable to others
rw/r	Read-Write to user, read to others
rw/rw	Read-Write to all
r/r	Read to all
r	Read to user, inaccessable to others

Onscreen Help Information

```
PAWS? help cmd=permit
```

Example 1

In this example the permissions for file lpitest.7 are changed to allow other users to read that file.

```
PAWS? permit file=lpitest.7 access=r/r
```

Example 2

In this example the permissions for file lpitest.8 are changed to allow the originating user read/write access while other users have only read.

```
PAWS? permit file=lpitest.8 access=rw/r
```

3.2.6.2.1.7 PRINT

The printer port cannot be accessed directly in PAWSWORK. If a hard copy is needed, the user may activate the printer port as described in the RTUSER1 Section where the use of the printer port is described: paragraph 3.2.5.2.

3.2.6.2.1.8 PURGE

Command Description

The PURGE command will accept only a source filename with a path to the user directory or to the FDD. The name may contain one or more wildcard characters (*). The file(s) which match the specified name in all character positions which do not contain * characters will be irreversibly deleted from the directory. When more than one file is to be purged, the user will be required to confirm the removal of each file by entering "y". Disk space occupied by the file(s) will be released.

Command Syntax

```
purge file=path1
```

where the parameters are defined as follows:

path1 - the source filename to be purged.

Onscreen Help Information

```
PAWS? help cmd=purge
```

Example 1

This example will erase the file test1 from the pawswork directory on the RDD.

```
PAWS? purge srce=test1
```

Example 2

This example will erase a file named test2 from the FDD.

```
PAWS?  purge file=/fd/test2
```

Example 3

This example will erase all files beginning with test from the RDD.

```
PAWS? purge file=test*
```

3.2.6.2.2 PAWSWORK Control Commands

There are four PAWSWORK Control Commands which allow the user to exit PAWSWORK, configure the system hardware, resume PAWSWORK activities and to get onscreen help. The use of these commands will be discussed in this section.

3.2.6.2.2.1 BYE

Command Description

The BYE command will end the PAWS Controller Software session and return the user to the "PAWS login" prompt. This command must be executed to exit PAWSWORK and "shutdown" entered at the PAWS login prior to powering down the unit.

Command Syntax

```
bye
```

Onscreen Help Information

```
PAWS? help cmd=bye
```

Example

```
PAWS? bye
```

3.2.6.2.2.2 CONFIGURE

Command Description

The CONFIGURE command allows the user to establish a system configuration for their PAWS session. Since the stand alone TU only supports the MTS function the user is recommended to use config std=9. Typing in "config" with no parameters will allow the user to edit the current configuration. The VI editor must be used to make these changes. See Appendix E.

Note: In order to function properly, the hardware must be configured the same way as the file suggests. Otherwise, the PAWS Controller Software functions will not be looking in the correct place for the interface module, and no communication will be made to the ASP.

Command Syntax

```
config[ure] [std=stdconf] [op=operation]
```

where the parameters are defined as follows:

stdconf - A pre-established standard configuration identifier.

operation - The req[uest] operation is the default.

Onscreen Help Information

```
PAWS? help cmd=config
```

Example

This example sets the standard configuration to 9 which is the standalone MTS function.

```
PAWS? config std=9
```

3.2.6.2.2.3 HELP

Command Description

The HELP command provides the user with help information regarding the Operator Interface commands.

Command Syntax

```
help cmd=xxx
```

where xxx is the name of a PAWS command.

Onscreen Help Information

```
PAWS? help
```

Example 1

This example will display a syntax summary of all the Operator Interface Commands.

```
PAWS? help
```

Example 2

This example will display help information for the modify command.

```
PAWS? help cmd=mod
```

3.2.6.2.2.4 Resume

Command Description:

The resume command shall allow the user to return to a previously SUSPENDed process and pick up where it left off. The resume command will restore terminal control to the process that was last SUSPENDed.

Command Syntax:

```
resume
```

Onscreen Help Information

```
PAWS? help cmd=resume
```

3.2.6.2.3. Global Commands

The Global Commands allow the user to set, list and clear global parameters to be used during a PAWSWORK session. When operating in a standalone MTS configuration, these commands are not used but will be used extensively when using future PAWS Controller Software functions.

3.2.6.2.3.1 CGLOBAL

Command Description

The CGLOBAL command clears global parameters which had previously been set by the set global parameter command. Any use of a command which requires the deleted parameter will cause an error unless the deleted parameter is respecified. Any use of a command for which that the deleted parameter is optional will utilize the default value specified for that command. If no parameters are specified, all global definitions will be eliminated.

Command Syntax

```
cg[lobal] [p1][p2]...[pn]
```

Where the parameters are defined as follows:

p1, p2, p2 - Optional parameters which define the name of a previously set global parameter.

Onscreen Help Information

```
PAWS? help cmd=cg[lobal]
```

Example

In this example, all parameters previously set by set global will be deleted.

```
PAWS? cg
```

3.2.6.2.3.2 LGLOBAL

Command Description

The LGLOBAL command lists global parameters and their values for the previously set global parameters.

Command Syntax

```
lg[lobal] [p1][p2]...[pn]
```

Where the parameters are defined as follows:

p1, p2, pn - Optional parameters which define the name of a previously set global parameter.

Onscreen Help Information

```
PAWS? help cmd=lg
```

Example 1

In this example, all previously defined parameters are displayed.

```
PAWS? lg
```

Example 2

In this example, the global parameters for id and xlat are displayed.

```
PAWS? lg id xlat
```

3.2.6.2.3.3 SGLOBAL

Command Description

The SGLOBAL command allows the user to set global parameter value(s). The global values set by this command will be used as the default VALUE for the parameter by all of the procedures entered after the SGLOBAL command.

Command Syntax

```
sg[lobal] [parameter keyword 1]=[parameter value 1] [parameter
          keyword 2]=[parameter value 2] ...[parameter keyword
          n]= [parameter value n]
```

Where the parameters are defined as follows:

parameter keyword - Name for the global parameter being defined.

parameter value - Value assigned to the parameter keyword.

Onscreen Help Information

```
PAWS? help cmd=sg
```

Example 1

This example sets the global default id to the value kk.

```
PAWS? sg id=kk
```

Example 2

This example sets the translator specification (i.e., xlat) to "c" and sets the history listing parameter (i.e., hist) to "no".

```
PAWS? sg xlat=c, hist=no
```

3.2.6.2.4. PAWS Controller Software

The PAWS Controller Software provides the user with several ASP related functions. In the stand alone TU configuration only the MTS function is available. In the future the user will be able to select from several additional software functions.

3.2.6.2.4.1 MTS

Command Description

The MTS command invokes the MTS software function for controlling, inspecting and changing memory of the ASP. When the MTS function is invoked, the MTS software is initialized, and the MTS display, shown in Figure 3-8, is displayed to the user. The MTS Function supports three operational modes: SPL, CP and AP. The display shown in Figure 3-8 supports both the SPL and CP modes. For the AP mode, a slightly different display is used as shown in Figure 3-9. When the AP mode of operation is invoked, the CSAR and CTL WORD fields will be cleared to reflect AP operation as reflected by the MAR and Arithmetic Element Control Word (AECW) fields.

Section 3.3 will describe the full use of this function.

Command Syntax

```
mts
```

This command has no parameters.

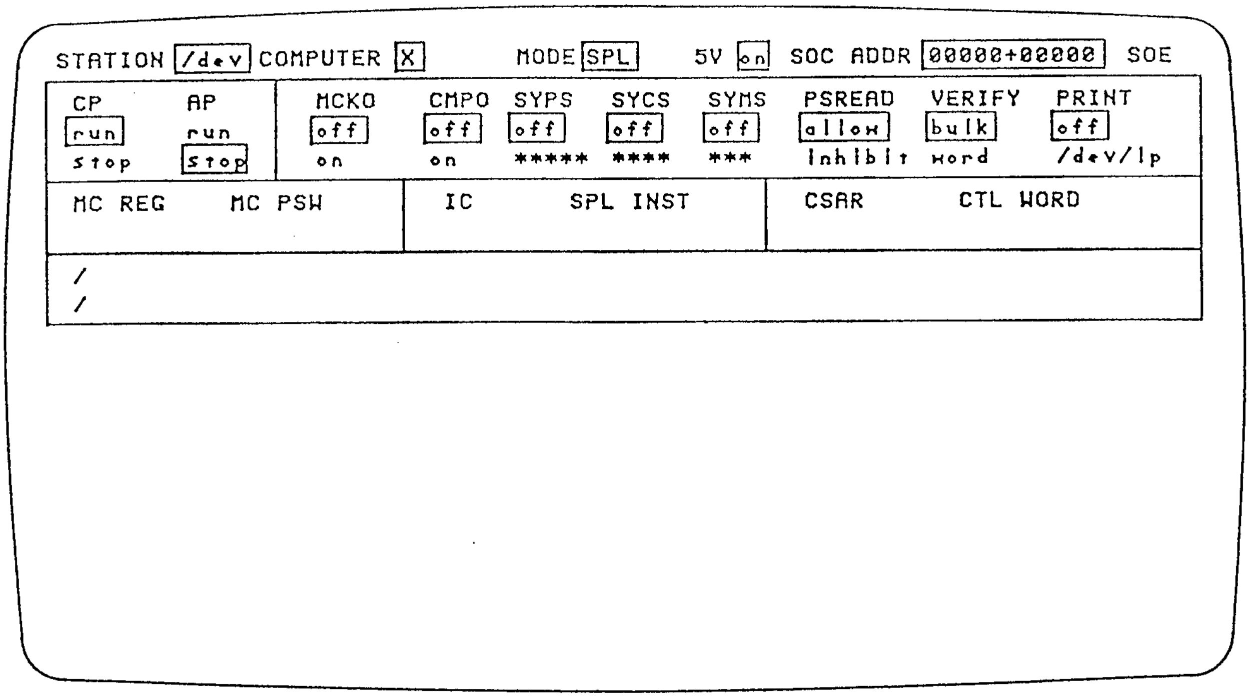
Onscreen Help Information

```
PAWS? help cmd=mts
```

Example

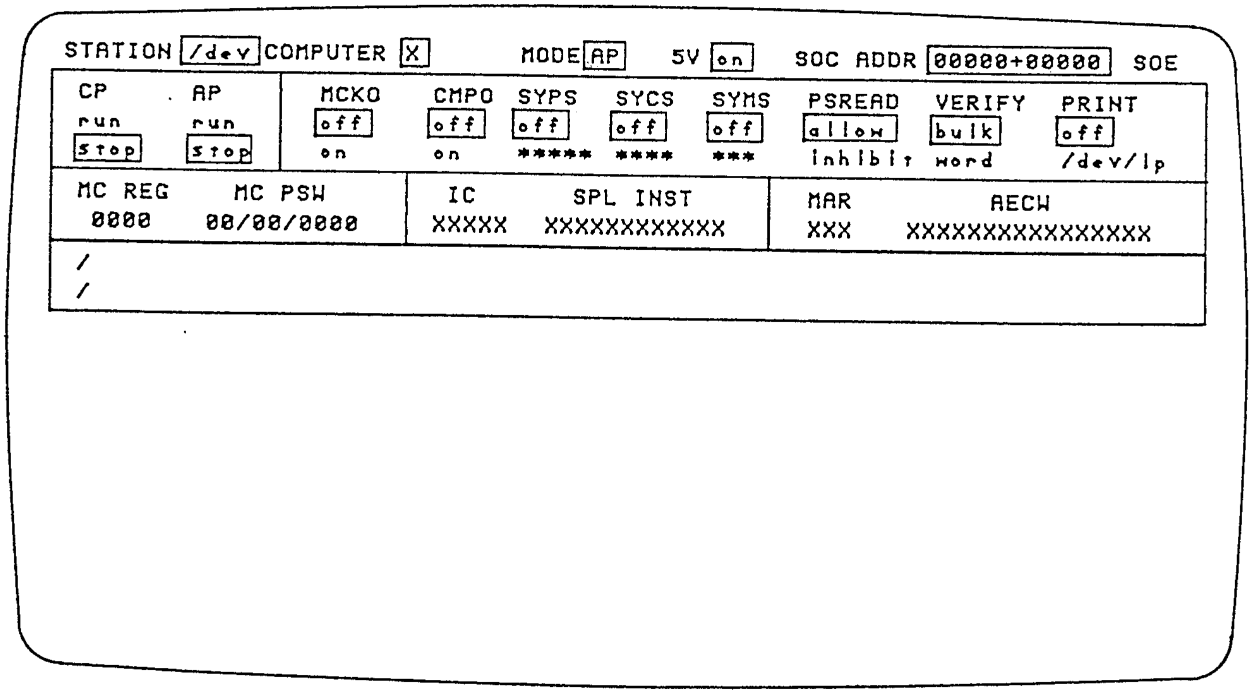
In this example the MTS function will be invoked. A standard configuration must be entered prior to invoking this function.

```
PAWS? mts
```



```
STATION /dev COMPUTER X     MODE SPL   5V on  SOC ADDR 00000+00000  SOE
 CP       AP    MCKO   CMPO  SYPS   SYCS  SYMS  PSREAD   VERIFY  PRINT
 run      run   off    off   off    off   off   allow    bulk    off
 stop     stop  on     on    *****  ****  ***   inhibit  word    /dev/lp
 MC REG     MC PSW   IC      SPL INST       CSAR       CTL WORD

 /
 /
```

Figure 3-8. MTS Display for SPL and CP Mode



```
STATION /dev COMPUTER X     MODE AP    5V on  SOC ADDR 00000+00000  SOE
 CP       AP    MCKO   CMPO  SYPS   SYCS  SYMS  PSREAD   VERIFY  PRINT
 run      run   off    off   off    off   off   allow    bulk    off
 stop     stop  on     on    *****  ****  ***   inhibit  word    /dev/lp
 MC REG    MC PSW       IC     SPL INST      MAR        AECW
  0000    00/00/0000    XXXXX  XXXXXXXXXXXX  XXX   XXXXXXXXXXXXXXXX
 /
 /
```

Figure 3-9. MTS Display for AP Mode

3.3 MTS Function Operation

3.3.1 Introduction to the MTS Screen

The screen display for the MTS function is shown in Figure 3-8. This display is divided into four areas. The top area of the MTS display provides the user with Status and Control information. The second area, is the User Command Entry Field. The third area is the Data Display Field and the fourth area is the Error Message Field. The MTS default information is highlighted in the figures by drawing a rectangle around the selectable information. On the real TU display, this information would be presented as reverse video. Also, the X's shown in the Register subfields represents displayed hexidecimal data that changes during MTS operation.

3.3.1.1 Status and Control Field

The Status and Control Field provides the user with information on the present state of the MTS software and the ASP. Initially the display reflects that to be used for Control Processor (CP) or Signal Processing Language (SPL) modes of operation as designated by the Control Store Address Register (CSAR) and Control (CTL) Word fields. The following paragraphs will summarize the various subfields of the status and control field.

Status/Control Subfields

Station Subfield - The Station subfield contains the device name of the terminal that the operator is logged into. This subfield will be /dev by default.

Computer Subfield - The Computer subfield informs the operator which unit of the ASP is being operated. This subfield will display /dev by default.

Mode Subfield - The Mode subfield informs the operator which computer mode is being operated: SPL, CP or AP.

5V Subfield - The 5V subfield informs the operator whether the 5V from the ASP is present.

SOC Address Subfield - When a Stop On Compare (SOC) address is set, this subfield will display that address and SOC will blink when SOC is enabled.

SOE Subfield - This subfield will blink when Stop On Error (SOE) is enabled by the operator.

CP Subfield - This subfield displays the status of the CP. The CP may be in the RUN or STOP state. When first invoked, the CP will be in the RUN state. The CP may be stopped using the STOP command and restarted with the START command.

Arithmetic Processor (AP) Subfield - This subfield displays the status of the AP. The AP may be in the RUN or STOP state. When first invoked, the AP will be in the STOP state. The AP may be started using the START command and stopped with the STOP command.

Machine Check Overide (MCKO) Subfield - This subfield identifies the present status of the MCKO. The state of this subfield may be changed with the OVERRIDE command.

Compare Micro Program Overide (CMPO) Subfield - This subfield identifies the present status of the CMPO. The state of this subfield may be changed with the OVERRIDE command.

Sync Program Store (SYPS) Subfield - This subfield identifies the state of the PS Sync. The initial state is set to off. The PS Sync may be enabled with the SYNC command. When enabled, the sync address will be displayed in this subfield.

Sync Control Store (SYCS) Subfield - This subfield identifies the state of the CS Sync. The initial state is set to off. The CS Sync may be enabled with the SYNC command. When enabled, the sync address will be displayed in this subfield.

Sync Micro Store (SYMS) Subfield - This subfield identifies the state of the MS Sync. The initial state is set to off. The MS Sync may be enabled with the SYNC command. When enabled, the sync address will be displayed in this subfield.

PSREAD Subfield - This subfield identifies the state of the PSREAD switch. When initialized, PSREAD will be ALLOWED. When PSREAD is enabled with the PSREAD command, the information in the Register Subfields will be displayed when the CP is stopped.

VERIFY Subfield - This subfield is controlled by the VERIFY TYPE command. When initialized, the Verify subfield will be set to BULK. This means that when writing to the ASP, the entire block written will be read back and verified. When changed to WORD, each word written to the ASP will be read back and verified.

PRINT Subfield - When PRINT is ON output written to the data display field will also be written to the associated print file. When PRINT is OFF, data will only be written to the Data Display Field. The PRINT subfield may be changed using the Define Printer and PRINT command.

Register Subfields

Machine Check (MC) Register (REG) Subfield - When PSREAD is ALLOWED, this subfield will display the MC Register when a machine check is detected when a MTS command is executed. When PSREAD is ALLOWED and CPSTEAL is enabled, this subfield will be updated once per second.

MC Program Status Word (PSW) Subfield - This subfield will display MC PSW. When PSREAD is ALLOWED, the MC PSW will be updated when a machine check is detected. If PSREAD is ALLOWED and CPSTEAL is enabled, this subfield will be updated once per second.

Instruction Counter (IC) Subfield - When PSREAD is ALLOWED, this subfield will display the IC Register when the CP has stopped. If PSREAD is ALLOWED and CPSTEAL is enabled, this subfield will be updated once per second.

SPL INST Subfield - This subfield will display the SPL Instruction as designated by the IC Register when PSREAD is ALLOWED, and the CP is stopped. When PSREAD is ALLOWED and CPSTEAL is enabled, this subfield will be updated once per second.

CP Operational Mode (See Figure 3-8)

Control Store Address Register (CSAR) Subfield - When in SPL or CP mode and PSREAD ALLOWED, this subfield will display the CSAR when the CP has stopped. If PSREAD is ALLOWED and CPSTEAL is enabled, this subfield will be updated once per second.

Control (CTL) WORD Subfield - This subfield will display the Control Word (CW) located at the address displayed in the CSAR when PSREAD is ALLOWED and the CP is stopped. When PSREAD is ALLOWED and CPSTEAL is enabled, this subfield will display dashes.

AP Operational Mode (See Figure 3-9)

Micro Store Address Register (MAR) Subfield - When in the AP mode and PSREAD is ALLOWED, this subfield will display the contents of the MAR when the CP is stopped. If PSREAD is ALLOWED and CPSTEAL is enabled, this subfield will be updated once per second.

Arithmetic Element Control Word (AECW) Subfield - This subfield will display the AECW located at the address displayed by the MAR when PSREAD is ALLOWED and the CP is stopped. When PSREAD is ALLOWED and CPSTEAL is enabled, this subfield will be updated once per second.

3.3.1.2 User Command Entry Field

The user may enter a MTS command using one of two techniques. The first approach is in the form of a text line entered in the User Command Entry Field terminated by a carriage return <CR>.

The second approach uses the keyboard numeric and function keys to execute commands that do not require parameters. These keyboard entries are not displayed on the screen and are processed as soon as they are recognized.

3.3.1.2.1 User Command Character Entry

The User Command Entry Field consists of two lines: previous command and present command line. The user may enter a command at the cursor which flashes at the start of the second of the two lines, i.e. the present command line. Those characters entered on the present command line will be parsed as commands by the MTS software and processed after the carriage return has been entered by the user. Once the command has been processed, the entered characters will be moved to the previous command line and the cursor will be reposition to the start of the second data entry line for the next user input. When the user enters a backspace character, the previous (non-function/non-cursor) key will be ignored. If a delete (DEL) character is entered by the user, all user entered characters on present command line will be erased and the cursor will be repositioned to the beginning of the second line. The result or response to the command will be displayed in the Status and Control Field, the Display Data Field or in the Error Message Field.

3.3.1.2.2 MTS Command Syntax

The form for the MTS command syntax is shown below:

```
(keyword) (sep) (param) (sep) ... (sep) (param)
```

where (keyword) is any valid command keyword or its abbreviation which will be defined in Section 3.3.3, (sep) is any combination of a single comma and one or more blanks, and (param) is any string of characters (including the null string) not including commas or blanks.

Entering a Command Keyword

The user must enter a valid keyword on the present command line. The MTS keywords may be accessed by using the MTS Onscreen Help Function by entering "help" on the present command line. If the keyword is not a valid command, an error will be displayed in the Error Message Field and the present command line will be cleared.

Any command entered on the present command line that requires that the CP be stopped in order to execute the entered command, the MTS function will (if the CP is running) first stop the CP, then perform the specified operation(s), then restart the CP at the rate in which it was running.

Entering a Parameter or a Parameter Keyword

If the parameter is an address, data or a count, the parameters are assumed to be entered in Hexadecimal format. This default may be overidden (by the user) on any parameter by enclosing the numeric quantity in parentheses preceded by the letter "O" for octal or "D" for decimal (i.e., D(139) to be decoded in decimal). Data parameters may be preceded by a unary minus sign and will then be decoded into a two's complement negative binary number. If the parameter can be decoded in the applicable radix, the binary value will be returned; if it cannot be decoded or the value entered is not within a legal range, an error message will be displayed in the Error Message Field.

If the parameter is a file or device name, the parameters will be examined to determine if the string is a syntactically valid qualified file or device name. If invalid, an error will be displayed in the Error Message Field.

If the parameter is a parameter keyword, the parameters will be validated for each MTS command of this type. An example of a command which uses this type of parameter is the "mode cp" command. In this command entry, "mode" is the command keyword and "cp" is the parameter keyword since it is neither an address, data, count, nor filename. If the parameter keyword is invalid, an error will be displayed in the Error Message Field.

3.3.1.3 Data Display Field

The Data Display Field consists of 12 lines (lines 13 through 24) which are dedicated to the display of data retrieved from the ASP. The Data Display Field will be updated whenever a user entered (or repeated) command solicits a data output.

3.3.1.4 Error Message Field

The Error Message Field is located at the bottom of the MTS display. The error messages which may be displayed are summarized in Appendix C.

3.3.2 MTS Keyboard Definition

The TU keyboard is shown in Figure 3-10. It operates similar to any other keyboard; however, when MTS has been invoked, the Numeric and Function keys may be used to enter MTS Commands which do not have parameters associated with the command. To use these keys the user must first press the "ESC" key followed by the function or numeric key. Either the numeric keys on the keypad or the keys accross the top of the keyboard may be used. When these keys are used, no charaters will appear in the User Entry Field. These added key functions and their associated MTS command are listed below. The definition of the MTS commands listed below are defined in Section 3.3.3.

ESC 1 - REPEAT

ESC 2 - HOME/WATCH

ESC 3 - PRINT

ESC 4 - CPSTEAL

ESC 5 - LOAD_DISPLAY

ESC 6 - RESET

ESC 7 - NEXT

ESC 8 - END

ESC 9 - SUSPEND

ESC 0 - Not Used

ESC F1 - +SCROLL

ESC F2 - -SCROLL

ESC F3 - PRINT_SCR

ESC F4 - START

ESC F5 - START_SOC

ESC F6 - START_SOE

ESC F7 - STOP/STEP

ESC F8 - Not Used

ESC F9 - Not Used

ESC F10 - Not Used

ESC J - MODIFY DISPLAY
After ESC J has been hit, Keys H, J, K, and L will serve as the cursor left, down, up and right, respectively.

DEL - INTERRUPT

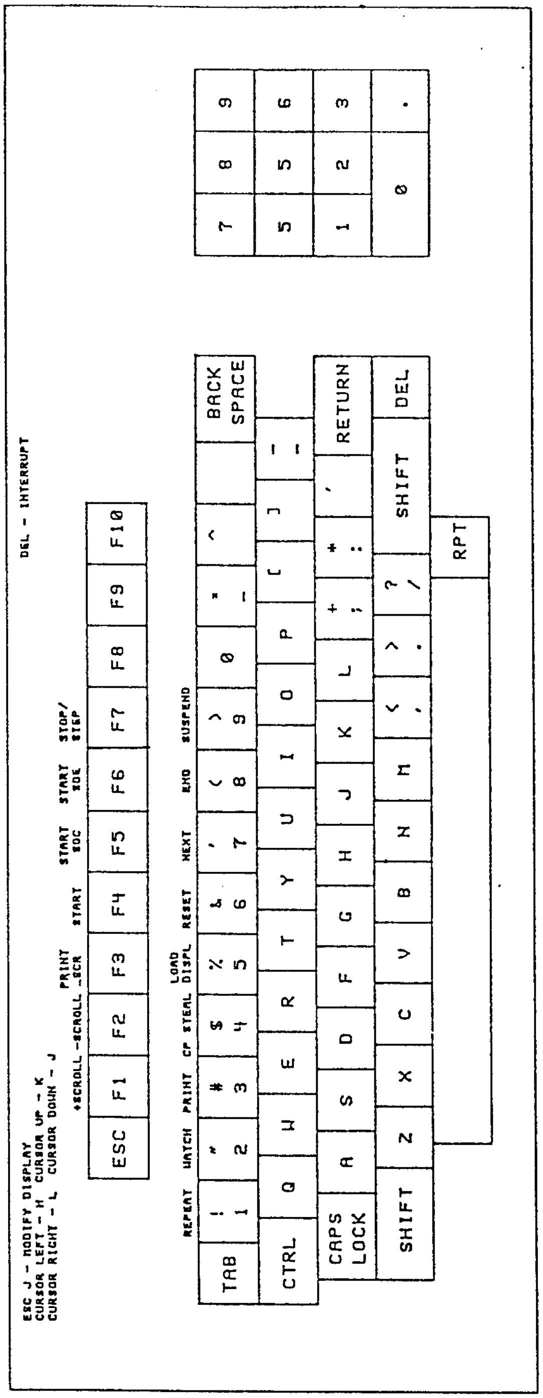


Figure 3-10. TU Keyboard Layout

3.3.3 MTS Command Definition

The three categories of MTS commands will be described in this section as follows:

1. MTS Support Commands

2. ASP Status and Control Commands

3. ASP Memory/Register Commands

These commands are summarized in Appendix B for quick reference.

During the discussion of the MTS commands in the following paragraphs, references will be made to foreground and background mode. This refers to a feature of the UNIX operating system which allows multiple processes to be operating at the same time; however, only one may be displayed. The process on the screen of the TU is considered to be in the foreground mode while the other process is operating in the background mode. In the stand alone MTS configuration, only the MTS function is available and therefore will always be in the foreground. In the future, addition ASP related software functions will be added which will require the foreground/background capabilities of UNIX.

3.3.3.1 MTS Support Commands

The MTS support commands are provided to help the user with the MTS session. These commands can be executed without the ASP connected. The syntax for the various commands will require mandatory parmeters identified by closed parenthesis (). Optional parameters are defined by square brackets[].

3.3.3.1.1 COMMENT

Command Description

The COMMENT command will allow comments to be put in include files generated with the PAWSWORK File Utilities (i.e., INCLUDE command)or print files (i.e., DEFINE PRINTER command) to assist with documentation of an MTS session. The user should type "* " or "comment" to activate this command.

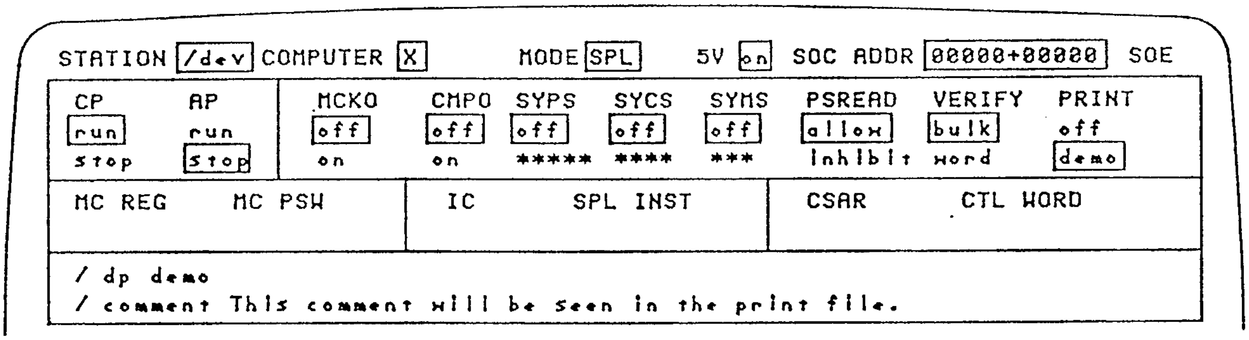
Command Syntax

```
comment (dt)
```

where the parameters are defined as follows:

dt - data and/or text

Example



```
STATION /dev COMPUTER X        MODE SPL    5V on  SOC ADDR 00000+00000  SOE

CP        AP       MCKO     CMPO   SYPS    SYCS   SYMS   PSREAD   VERIFY  PRINT
run       run      off      off    off     off    off    allow    bulk    off
stop      stop     on       on     *****   ****   ***    inhibit  word    demo

MC REG     MC PSW       IC         SPL INST          CSAR        CTL WORD

/ dp demo
/ comment This comment will be seen in the print file.
```

In this example, a print file named "demo" was created and the print function enabled. The comment "This comment will be seen in the print file" will be added to the "demo" print file.

3.3.3.1.2 ECP41

Command Description

The ECP41 command will inform the MTS software that an ASP with the extended Program Store memory is under test. The MTS software will increase the Program Store memory address range to account for the increased address space in an ASP with an increased Program Store of 256k. There are no changes to the display or user prompt when this command has been executed.

Command Syntax

```
ecp41
```

This command does not require any parameter definition.

3.3.3.1.3 END

Command Description

The END command will cause the MTS task to terminate. The ASP(s) will be left in the current state. If one of the SYNC functions had been enabled and still monitoring when the END command is entered, this SYNC functions will also be terminated. Display control will be passed to the next task waiting for foreground use of the display.

The END command may also be activated using the numeric/function key <esc>8.

Command Syntax

```
end
```

This command does not require any parameter definition.

3.3.3.1.4 HELP

Command Description

The HELP command will provide the user with on-line information about all the MTS commands. This command will provide a menu of all help available. The HELP files available for the individual commands will provide a command description, command syntax and an example of use.

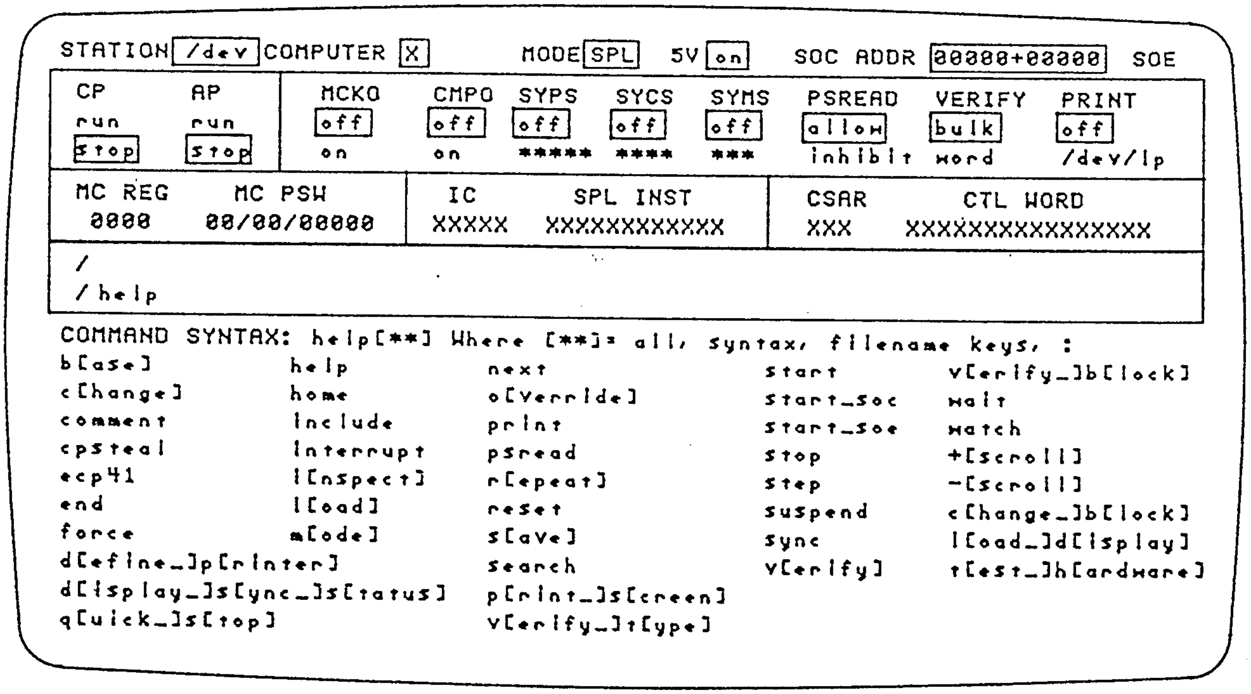
Command Syntax

help [keyword]

where the parameters are defined as follows:

keyword - MTS command abreviation.

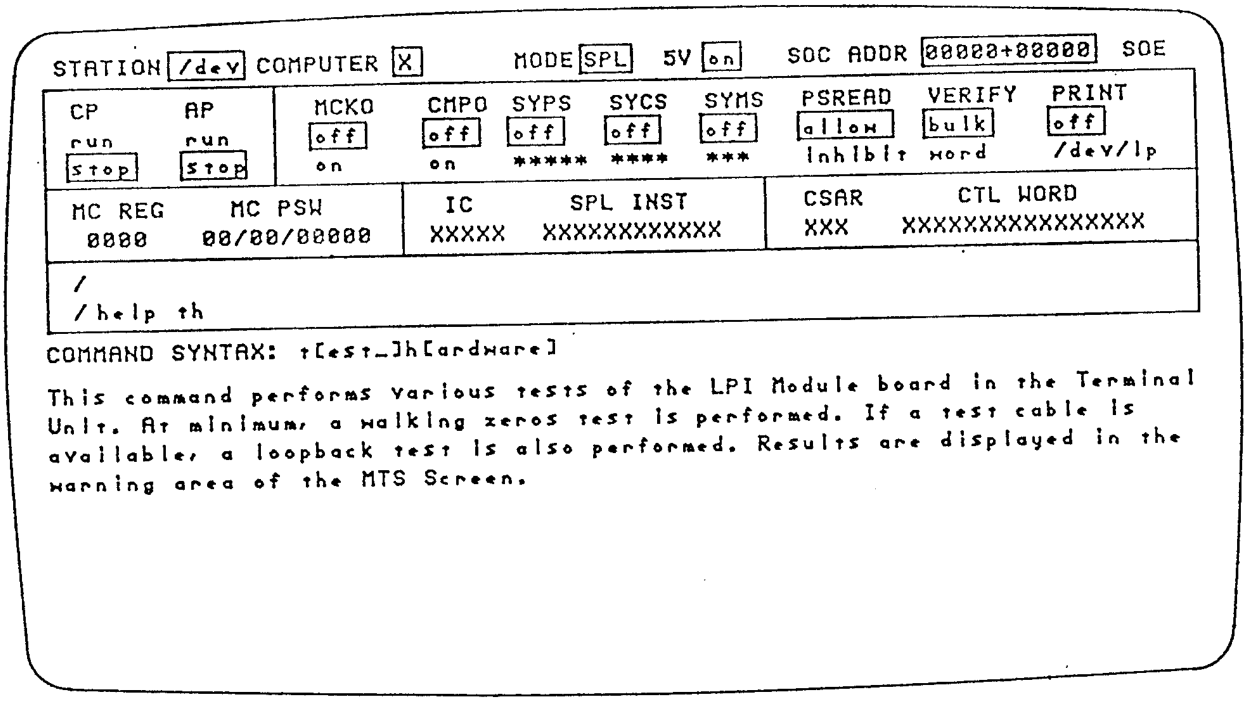
Example 1



```
STATION[/dev]COMPUTER [X]      MODE[SPL]  5V[on]   SOC ADDR [00000+00000]  SOE
 CP        AP       MCKO    CMPO  SYPS   SYCS   SYMS   PSREAD   VERIFY   PRINT
 run       run      [off]   [off] [off]  [off]  [off]  [allow]  [bulk]   [off]
[stop]    [stop]     on      on   *****  ****   ***    inhibit  word     /dev/lp
 MC REG    MC PSW          IC     SPL INST          CSAR     CTL WORD
  0000    00/00/00000      XXXXX  XXXXXXXXXXXX      XXX      XXXXXXXXXXXXXXXX
 /
 /help
COMMAND SYNTAX: help[**] Where [**]= all, syntax, filename keys, :
b[ase]          help            next            start           v[erify_]b[lock]
c[hange]        home            o[verride]      start_soc       halt
comment         include         print           start_soe       watch
cpsteal         interrupt       psread          stop            +[scroll]
ecp41           i[nspect]       r[epeat]        step            -[scroll]
end             l[oad]          reset           suspend         c[hange_]b[lock]
force           m[ode]          s[ave]          sync            l[oad_]d[isplay]
d[efine_]p[rinter]              search          v[erify]        t[est_]h[ardware]
d[isplay_]s[ync_]s[tatus]       p[rint_]s[creen]
q[uick_]s[top]                  v[erify_]t[ype]
```

In this example, the keyword for the MTS commands are displayed in the Data Display Field as shown above.

Example 2



```
STATION [/dev] COMPUTER [X]      MODE [SPL]  5V [on]   SOC ADDR [00000+00000]  SOE
CP        AP       MCKO    CMPO   SYPS    SYCS   SYMS   PSREAD    VERIFY   PRINT
run       run      [off]   [off]  [off]   [off]  [off]  [allow]   [bulk]   [off]
[stop]    [stop]   on      on     *****   ****   ***    inhibit   word     /dev/lp
MC REG    MC PSW        IC      SPL INST          CSAR     CTL WORD
0000      00/00/00000   XXXXX   XXXXXXXXXXXX      XXX      XXXXXXXXXXXXXXXX
/
/help th
COMMAND SYNTAX: t[est_]h[ardware]
This command performs various tests of the LPI Module board in the Terminal
Unit. At minimum, a walking zeros test is performed. If a test cable is
available, a loopback test is also performed. Results are displayed in the
warning area of the MTS Screen.
```

This example displays help information for the Test Hardware command as shown above.

3.3.3.1.5 HOME

Command Description

This command will cause the cursor to return to the next available position on the present command line. The HOME command will return the cursor to the next available input line even if the cursor has been moved to the Data Display Field using the MODIFY DISPLAY numeric/function key: <esc>j.

Command Syntax

This command will only be activated via the numeric/function key <esc>2.

3.3.3.1.6 INCLUDE

Command Description

This MTS INCLUDE command will access the file specified by the "filename" parameter of the command. The MTS software will then begin reading user commands from this file and processing them until either no commands remain on the file or the user presses the DEL key. As the commands are processed they are displayed in the User Command Entry Field as if they had been entered from the keyboard. The file may contain another INCLUDE command, but no further commands will be processed from the original file (i.e., INCLUDEs are not nestable).

In order to use this command, the user must first create a file using the PAWSWORK File Utilities described in the PAWSWORK Section of this document: Section 3.2.6.

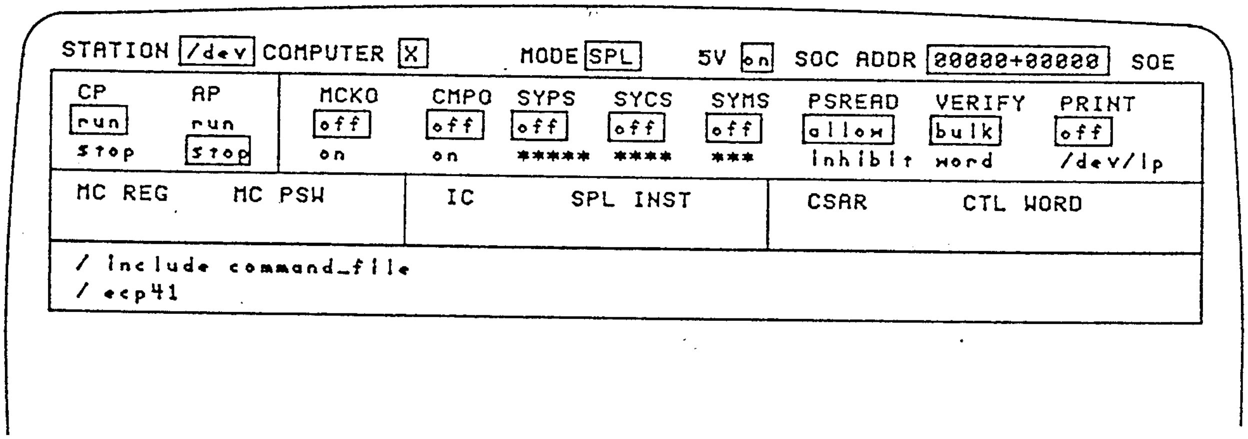
Command Syntax

```
include (fn)
```

where the parameters are defined as follows:

fn - file name

Example



```
STATION [/dev] COMPUTER [X]     MODE [SPL]    5V [on]  SOC ADDR [00000+00000]  SOE

CP       AP       MCKO    CMPO   SYPS    SYCS   SYMS   PSREAD    VERIFY   PRINT
[run]    run      [off]   [off]  [off]   [off]  [off]  [allow]   [bulk]   [off]
stop     [stop]   on      on     *****   ****   ***    inhibit   word     /dev/lp

MC REG     MC PSW     IC        SPL INST        CSAR        CTL WORD

/ include command_file
/ ecp41
```

In this example the file named "command_file" has been opened and the execution of the MTS files included in "command_file" has begun. The command presently being executed is "ecp41".

3.3.3.1.7 INTERRUPT

Command Description

The command will function only when another command is in progress. In this case, INTERRUPT will cause the command being processed to be terminated prematurely at the next reasonable point in processing. The INTERRUPT command must be entered by the DEL key.

Command Syntax

This command may only be activated via the DEL key.

3.3.3.1.8 NEXT

Command Description

This command will cause the present MTS foreground task to relinquish control of the MTS display and signal the next task waiting for foreground use of the display to assume control and initialize the terminal. The original task, MTS, will continue to run in background mode, monitoring the ASP and maintaining the internal states up to date. When the background task is subsequently signaled, it will once again assume control of the MTS display.

This command may also be activated using the numeric/function key <esc>7.

Command Syntax

```
next
```

This command does not require any parameter definition.

3.3.3.1.9 REPEAT

Command Description

The previous command entered will be repeated, once per second, the number of times specified by the count parameter or until the user presses the DEL key. If a negative count is specified, the repetition continues indefinitely and may be terminated only by the DEL key. Commands which may not be repeated are REPEAT, REPEAT_HARDWARE, INCLUDE, NEXT, INTERRUPT and END. Commands entered via numeric or function keys will not be repeated.

This command may also be activated using the numeric/function key <esc> 1. When activated using the numeric/function key or when no count is specified in the count parameter, the previous command will be repeated only once.

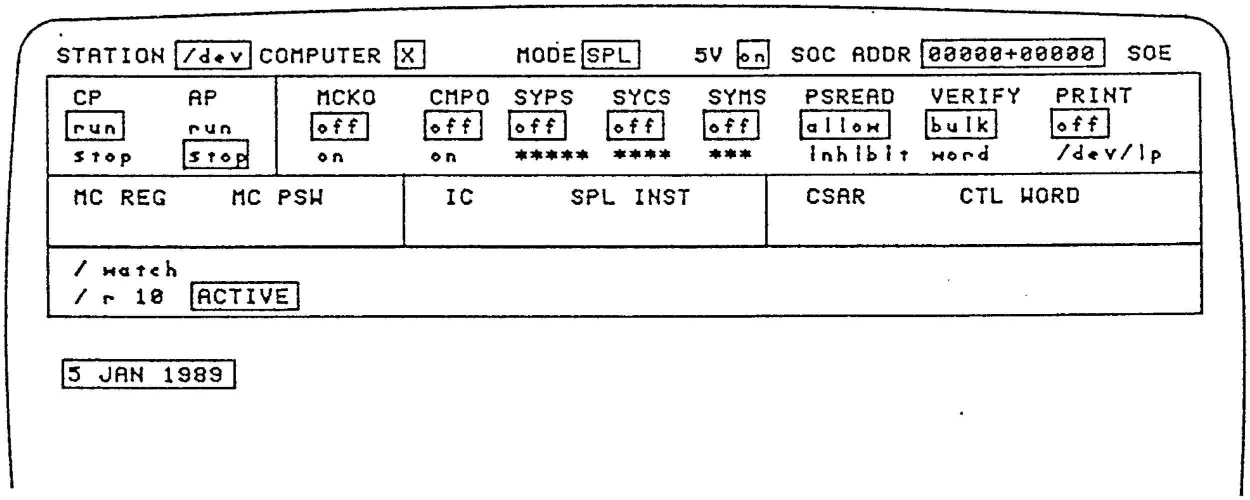
Command Syntax

```
r[epeat] [ct]
```

where the parameters are defined as follows:

ct - count

Example



```
STATION [/dev] COMPUTER [X]      MODE[SPL]    5V [on]  SOC ADDR [00000+00000]  SOE

CP       AP       MCKO    CMPO   SYPS    SYCS   SYMS   PSREAD   VERIFY  PRINT
[run]    run      [off]   [off]  [off]   [off]  [off]  [allow]  [bulk]  [off]
stop     [stop]   on      on     *****   ****   ***    inhibit  word    /dev/lp

MC REG      MC PSW      IC        SPL INST        CSAR        CTL WORD

/ watch
/ r 10 [ACTIVE]

[5 JAN 1989]
```

This example will repeat the execution of the WATCH command ten times. While the commands are being executed, an ACTIVE indicator is displayed on the present command line when the REPEAT command executes. When the WATCH command executes, a time/date indicator toggles in the Data Display Field.

3.3.3.1.10 REPEAT HARDWARE

Command Description

The REPEAT_HARDWARE command will cause the previous command to be repeated with continuous internal looping. If the optional R parameter is specified, a System Reset will be performed between repeated operations. The repetition rate will be a function of the Laboratory Panel Interface operation being performed. This rate will be high enough to permit laboratory test equipment diagnosis on the hardware. The commands which may be repeated are:

- INSPECT
- CHANGE
- FORCE

  If the CP is running when one of these 3 attempts a REPEAT HARDWARE, the CP is stopped and not restarted until looping terminates

- STOP
- START
- START SOC
- START SOE
- STEP
- RESET

The only way the user may stop the looping operation is via the DEL key. These commands will not write data to or update the MTS display while in the hardware repeat loop.

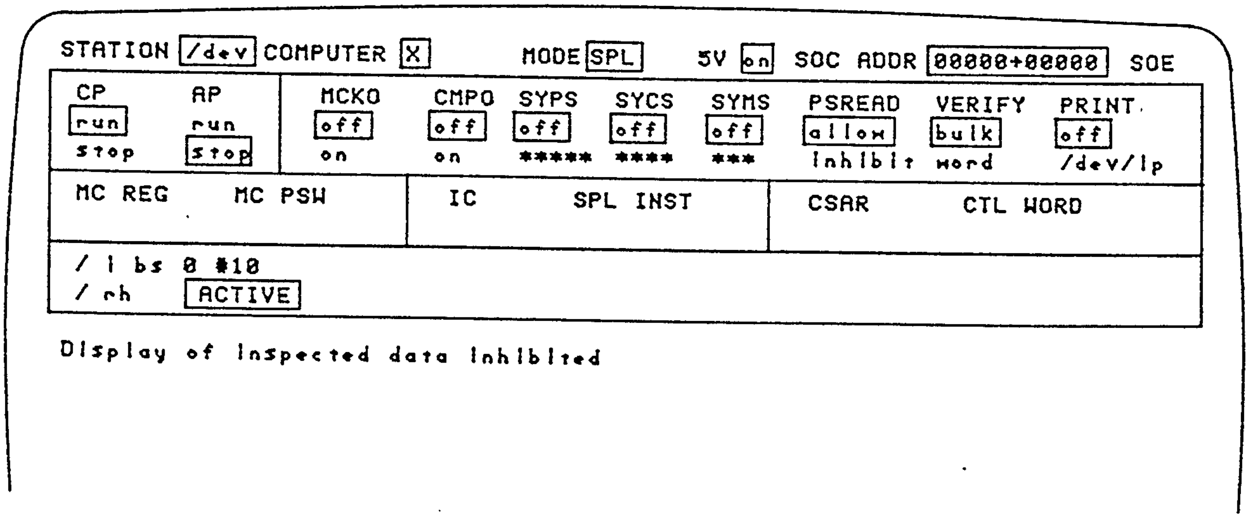
Command Syntax

```
r[epeat_]h[ardware] (r)
```

where the parameters are defined as follows:

r - perform system reset between repeats.

Example



```
STATION /dev COMPUTER X      MODE SPL   5V on  SOC ADDR 00000+00000  SOE
CP      AP     MCKO   CMPO  SYPS   SYCS  SYMS  PSREAD   VERIFY  PRINT
run     run    off    off   off    off   off   allow    bulk    off
stop    stop   on     on    *****  ****  ***   inhibit  word    /dev/lp
MC REG      MC PSW    IC        SPL INST         CSAR       CTL WORD
/ l bs 0 #10
/ rh      ACTIVE
Display of inspected data inhibited
```

This example will continuously send an Inspect Bulk Store command to the LPI Module. As the REPEAT HARDWARE command is executed, an ACTIVE indicator is displayed in the User Command Entry Field.

3.3.3.1.11 SCREEN

Command Description

The SCREEN command will enable or disable the display of information in the Data Display Field. The normal setting of this parameter will be "on", during which all command data will be written to the MTS display. If the screen parameter is set to off, the data from commands displaying results in the Data Display Field will not be shown on the screen. The MTS function will be initialized with the screen parameter set to "on".

Command Syntax

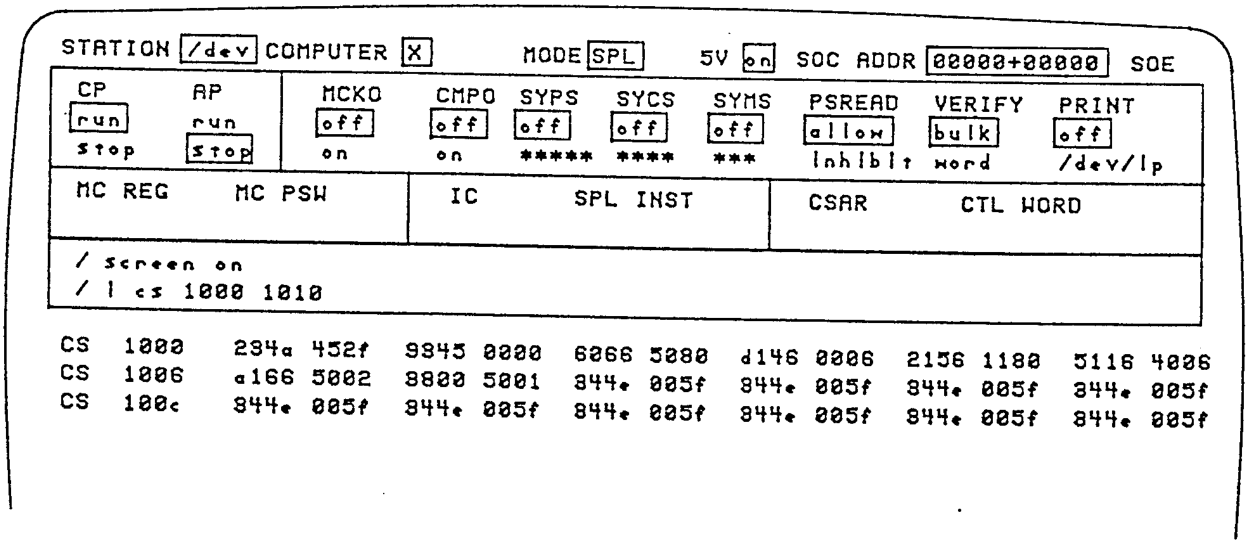
```
screen [nf]
```

where the parameters are defined as follows:

nf - "on" enables the Data Display Field

- "off" disables the Data Display Field

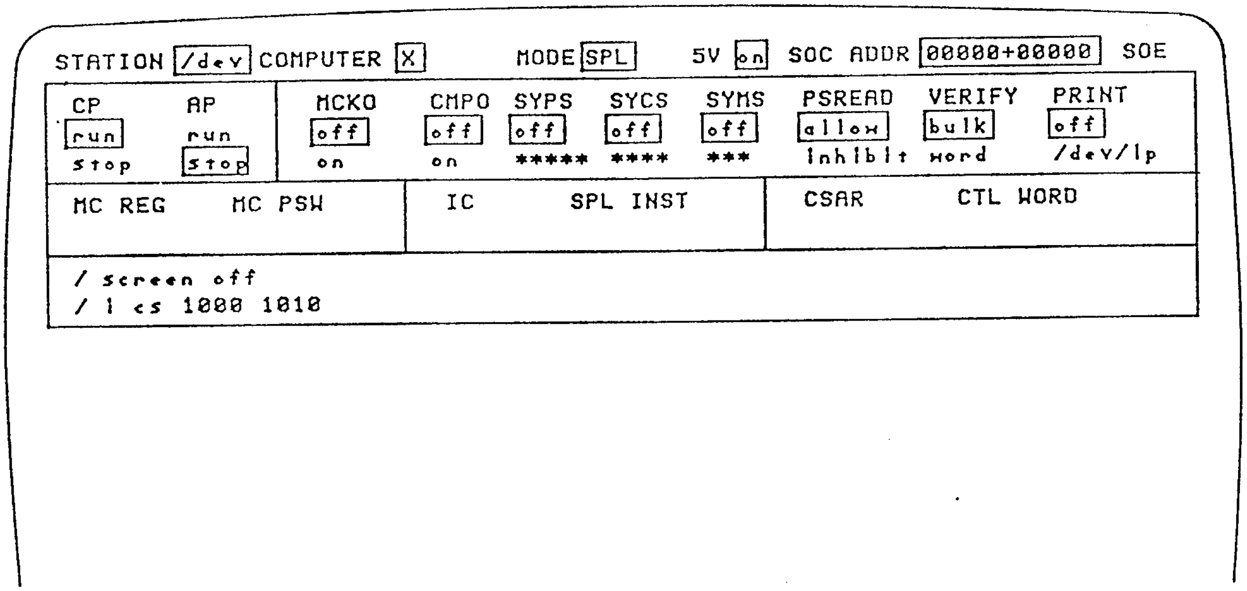
Example 1



```
STATION /dev COMPUTER X      MODE SPL    5V on  SOC ADDR 00000+00000  SOE
 CP        AP      MCKO    CMPO  SYPS   SYCS  SYMS  PSREAD   VERIFY  PRINT
 run       run     off     off   off    off   off   allow    bulk    off
 stop      stop    on      on    *****  ****  ***   inhibit  word    /dev/lp
 MC REG      MC PSW      IC       SPL INST       CSAR       CTL WORD

 / screen on
 / l cs 1000 1010

CS   1000   234a 452f   9845 0000   6066 5080   d146 0006   2156 1180   5116 4006
CS   1006   a166 5002   8800 5001   344e 005f   844e 005f   844e 005f   344e 005f
CS   100c   844e 005f   844e 005f   844e 005f   844e 005f   844e 005f   844e 005f
```

In this example, the data from bulk store would be displayed in the Data Display Field.

Example 2



```
STATION [/dev] COMPUTER [X]      MODE [SPL]   5V [on]  SOC ADDR [00000+00000]  SOE
CP       AP       MCKO    CMPO    SYPS    SYCS   SYMS   PSREAD    VERIFY   PRINT
[run]    run      [off]   [off]   [off]   [off]  [off]  [allow]   [bulk]   [off]
stop     [stop]   on      on      *****   ****   ***    inhibit   word     /dev/lp
MC REG     MC PSW      IC        SPL INST         CSAR        CTL WORD
/ screen off
/ 1 cs 1000 1010
```

In this example, the data from bulk store would NOT be displayed in the Data Display Field. There would be no change in the MTS display or user prompt when this command is executed.

3.3.3.1.12 SUSPEND

Command Description

The SUSPEND command will cause the foreground copy of the MTS Function to relinquish foreground use of the TU display, passing the foreground back to the PAWSWORK Operator Interface. The user may then initiate other subsystems, alter configuration, or execute file utilities. Foreground control of the terminal will return to the suspended MTS function through the "next" or "end" mechanism of other subfunctions initiated or when the user selects the RESUME function of the PAWSWORK Operator Interface.

This command may also be activated using the numeric/function key <esc> 9.

Command Syntax

```
suspend
```

This command does not require any parameter definition.

3.3.3.1.13 TEST HARDWARE

Command Description

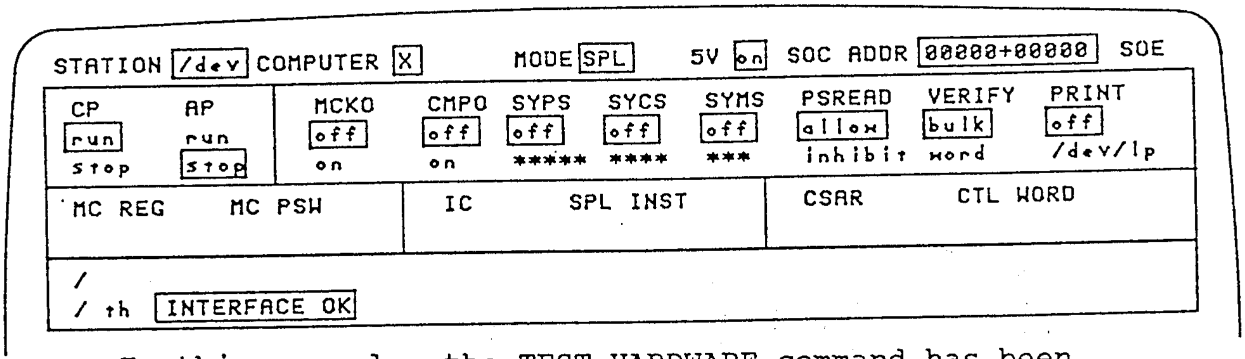
The TEST HARDWARE command will control the execution of BIT tests for the LPI module. Upon completion of the test, a message shall be returned to the operator indicating successful completion, or an error message will be generated.

Command Syntax

```
t[est_]h[hardware]
```

This command does not require any parameter definition.

Example



```
STATION [/dev] COMPUTER [X]      MODE [SPL]    5V [on]  SOC ADDR [00000+00000]  SOE
 CP        AP      MCKO    CMPO   SYPS    SYCS   SYMS   PSREAD   VERIFY  PRINT
[run]      run     [off]   [off]  [off]   [off]  [off]  [allow]  [bulk]  [off]
 stop     [stop]    on      on    *****   ****   ***    inhibit  word    /dev/lp
 MC REG     MC PSW       IC        SPL INST         CSAR        CTL WORD

 /
 / th [INTERFACE OK]
```

In this example, the TEST HARDWARE command has been executed and has passed, signified by the "INTERFACE OK" displayed in the User Entry Field.

3.3.3.1.14 WAIT

Command Description

The WAIT command will cause a delay for the specified number of seconds. When no parameter is specified, the command will wait until the appropriate processor (CP for SPL or CP modes, AP for AP mode) stops. If it is already stopped, the command will have no effect. In either case, no commands will be read or processed from the user (or from an included command file) during the delay; the delay may be prematurely terminated by the DEL key.

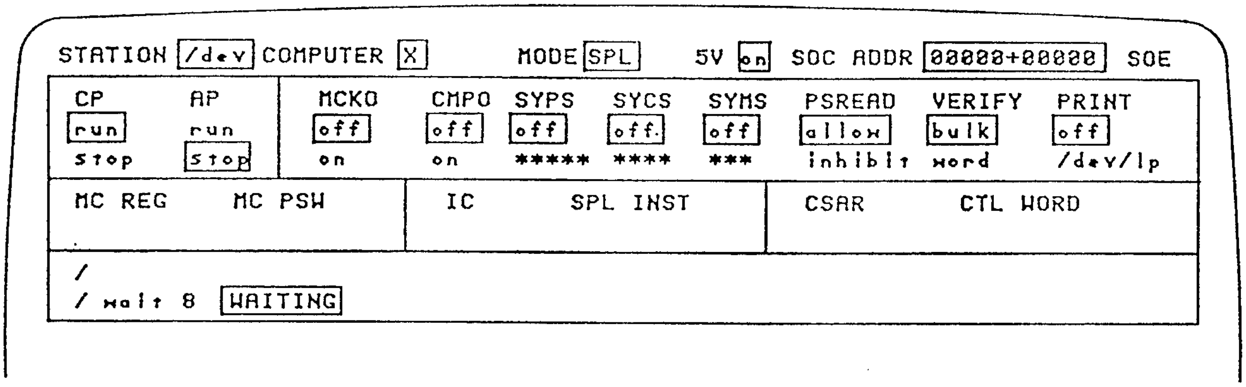
Command Syntax

```
wait [ct]
```

where the parameters are defined as follows:

ct - count

Example



```
STATION /dev COMPUTER X        MODE SPL    5V on  SOC ADDR 00000+00000  SOE
CP        AP      | MCKO    CMPO  SYPS   SYCS   SYMS  PSREAD   VERIFY  PRINT
run       run     | off     off   off    off    off   allow    bulk    off
stop      stop    | on      on    *****  ****   ***   inhibit  word    /dev/lp
MC REG      MC PSW    | IC        SPL INST    | CSAR        CTL WORD

/
/ wait 8 WAITING
```

In this example the MTS software will wait 3 seconds. During this wait period, a "WAITING" prompt is displayed in the User Entry Field.

3.3.3.1.15 WATCH

Command Description

The WATCH command will display the current date in the Data Display Field. If this command is executed a second time, the time will be displayed.

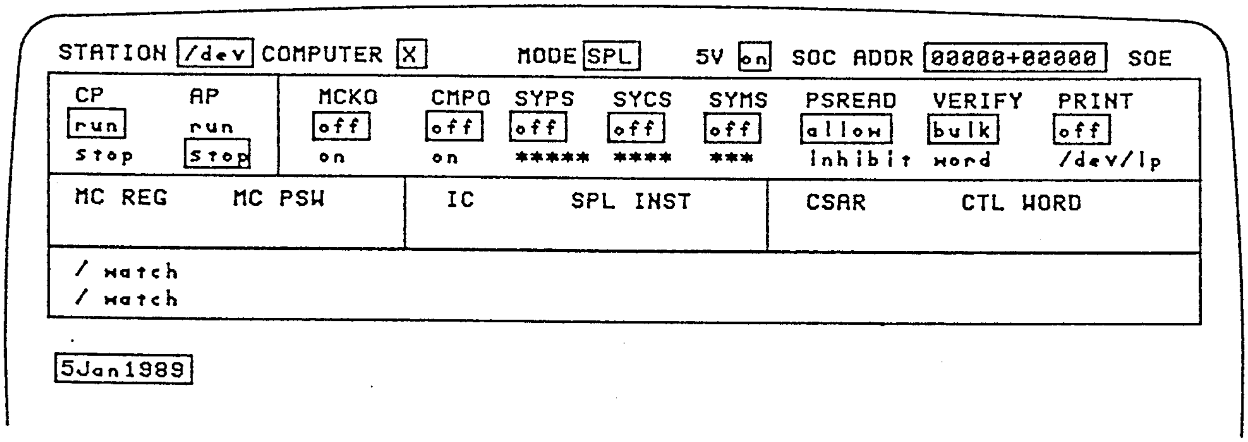
This command may also be activated using the numeric/function key <esc> 2.

Command Syntax

```
watch
```

This command does not require any parameter definition.

Example



```
STATION /dev COMPUTER X        MODE SPL    5V on  SOC ADDR 00000+00000  SOE
CP        AP      | MCKO    CMPO  SYPS   SYCS   SYMS  PSREAD   VERIFY  PRINT
run       run     | off     off   off    off    off   allow    bulk    off
stop      stop    | on      on    *****  ****   ***   inhibit  word    /dev/lp
MC REG      MC PSW    | IC        SPL INST    | CSAR        CTL WORD

/ watch
/ watch

5Jan1989
```

In this example, the WATCH has been executed to show the date.

3.3.3.2 ASP Control and Status Commands

The ASP Control and Status Commands are used to enable and disable various MTS software functions during a MTS session. The result of using these commands are displayed in the Control/Status Field of the MTS display as described in Section 3.3.1.1.

3.3.3.2.1 CPSTEAL

Command Description

The CPSTEAL command will toggle the cpsteal switch on and off. If the cpsteal switch is on (enabled), a small blinking diamond will be displayed just to the right of the CP "stop" status indicator. If PSREAD has also been Allowed the MTS Function will monitor the MC REG, MC PSW, IC, SPL INST and CSAR and dynamically update the Regsister Subfield at one second intervals even when the CP is running. This requires "stealing" the CP for a small portion of each second. If this is intolerable to the executing code in the ASP, cpsteal will be toggled off (disabled). The cpsteal switch will be initially off (disabled). When it is off, the Register subfields of the MTS display will be cleared whenever the CP is running.

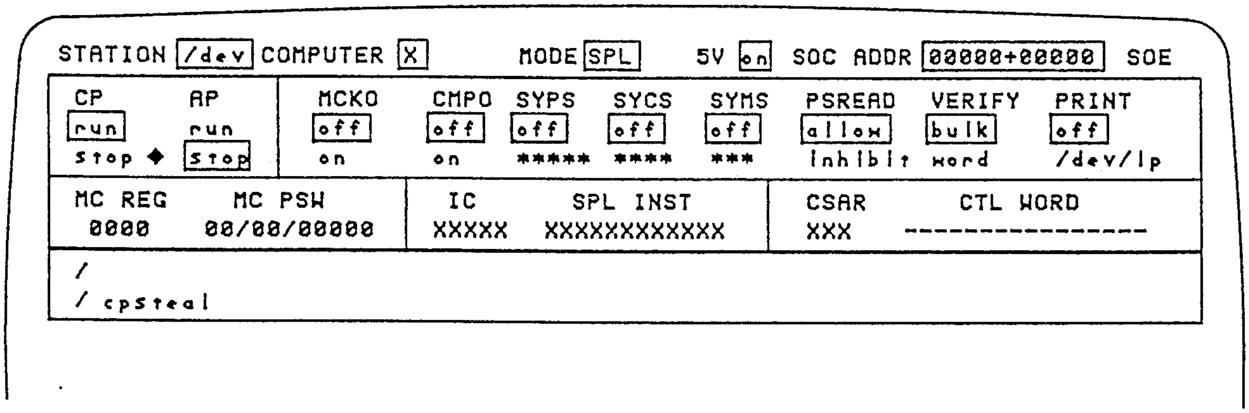
This command may also be activated using the numeric/function key <esc> 4.

Command Syntax

    cpsteal

This command does not require any parameter definition.

Example



```
STATION [/dev] COMPUTER [X]      MODE [SPL]    5V [on]  SOC ADDR [00000+00000]  SOE

CP        AP      MCKO   CMPO   SYPS   SYCS   SYMS   PSREAD    VERIFY   PRINT
[run]     run     [off]  [off]  [off]  [off]  [off]  [allow]   [bulk]   [off]
stop ◆   [stop]   on     on     *****  ****   ***    inhibit   word     /dev/lp

MC REG     MC PSW        IC      SPL INST        CSAR      CTL WORD
 0000    00/00/00000    XXXXX   XXXXXXXXXXXX     XXX     ----------------

/
/ cpsteal
```

In this example, the CP STEAL command is being executed as indicated by the flashing diamond next to the CP STOP Field.

3.3.3.2.2 DEFINE PRINTER

Command Description

The DEFINE PRINTER command will allow the user to define a print file for the MTS session. The name of the print file will be displayed in the Print subfield of the MTS display. All entries from the TU and responses from the ASP will be recorded in the defined print file for later examination or for printing of a hard copy.

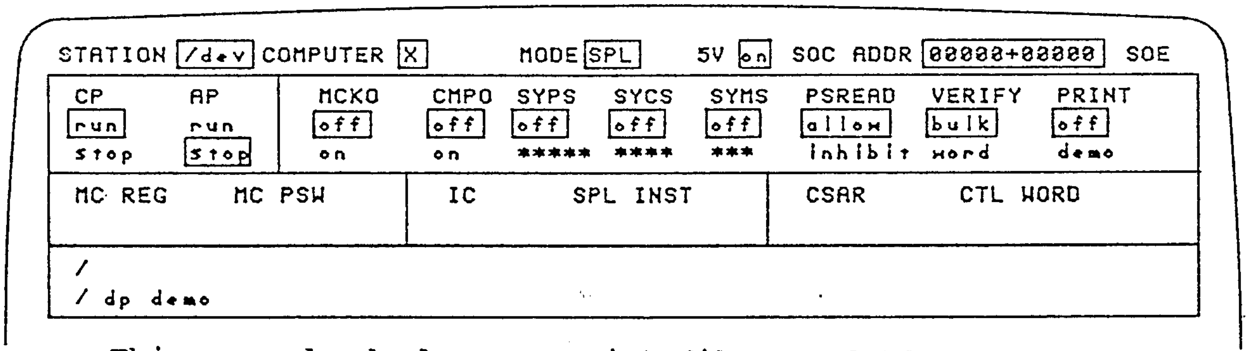
Command Syntax

```
d[efine_]p[rinter] (fn)
```

where the parameters are defined as follows:

fn - file name

Example



```
STATION /dev COMPUTER X       MODE SPL    5V on  SOC ADDR 00000+00000  SOE

CP       AP      MCKO    CMPO  SYPS   SYCS  SYMS  PSREAD   VERIFY  PRINT
run      run     off     off   off    off   off   allow    bulk    off
stop     stop    on      on    *****  ****  ***   inhibit  word    demo

MC REG    MC PSW    IC       SPL INST          CSAR      CTL WORD

/
/ dp demo
```

This example declares a print file named "demo". The print subfield displays the file name "demo"; however, the "off" indicates that the print has not been enabled. Print must be enabled with the PRINT command.

3.3.3.2.3 DISPLAY SYNC STATUS

Command Description

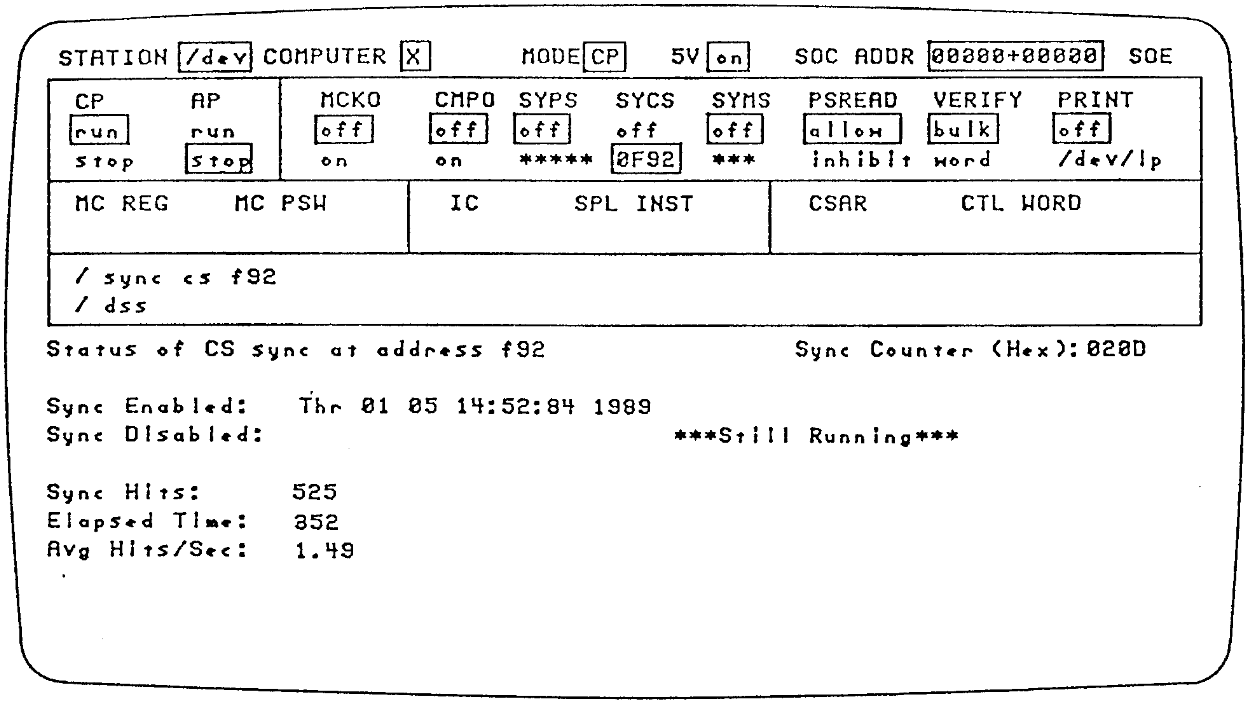
When SYNC command has been enabled, the SYNC will monitor the specified sync in the background mode. The operator may interrogate the SYNC status by entering the DISPLAY SYNC STATUS command. When queried, the accumulated statistics will be displayed in the Data Display Field of the MTS display.

Command Syntax

```
d[isplay_]s[ync_]s[tatus]
```

This command does not require any parameter definition.

Example



```
STATION /dev COMPUTER X       MODE CP    5V on    SOC ADDR 00000+00000   SOE
 CP        AP     MCKO     CMPO   SYPS   SYCS   SYMS   PSREAD   VERIFY   PRINT
 run       run    off      off    off    off    off    allow    bulk     off
 stop      stop   on       on     *****  0F92   ***    inhibit  word     /dev/lp
 MC REG     MC PSW      IC          SPL INST          CSAR          CTL WORD

 / sync cs f92
 / dss
Status of CS sync at address f92                  Sync Counter (Hex):020D

Sync Enabled:    Thr 01 05 14:52:84 1989
Sync Disabled:                             ***Still Running***

Sync Hits:       525
Elapsed Time:    352
Avg Hits/Sec:    1.49
```

In this example, Sync has been enabled and DISPLAY SYNC STATUS has been requested and the Sync information is displayed in the Data Display Field.

3.3.3.2.4 FORCE

Command Description

Either CW (CP Micro Control Word) or AECW (Arithmetic Element Control Word) will be user specified, as will the entire instruction word. The ASP will be requested via the LPI module to force execution of the specified instruction. The MTS function will not attempt to ensure that the instruction is valid. No user output will be displayed.

Command Syntax

```
force (nf) (in)
```

where the parameters are defined as follows:

nf - cw for Control Word
   - aecw for Arithmetic Element Control Word

in - an instruction word

3.3.3.2.5 MODE

Command Description

The MODE command allows the user to specify which ASP processor (SPL, CP or AP) will be used during the MTS session. This command will be used to determine the type of instruction to be stepped as a result of the STEP command (SPL instruction, Micro Control Word, or AECW), and the storage type (i.e., PS, CS or MS) when performing a Stop-On-Compare address applies. The MODE subfield in MTS Status and Control Field will be changed to reflect the mode selection; the SOC Address subfield will be set to the SOC address. The default Stop-On-Compare address will be zero.

Command Syntax

```
m[ode] (nf) [ca]
```

where the parameters are defined as follows:

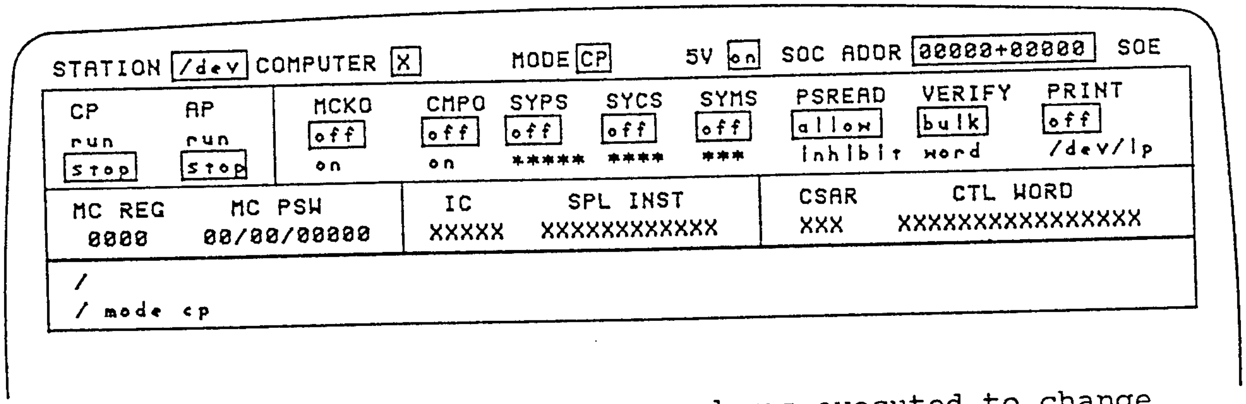
nf - spl for Signal Processing Language

- cp for Control Processor

- ap for Arithmetic Processor

ca - SOC address

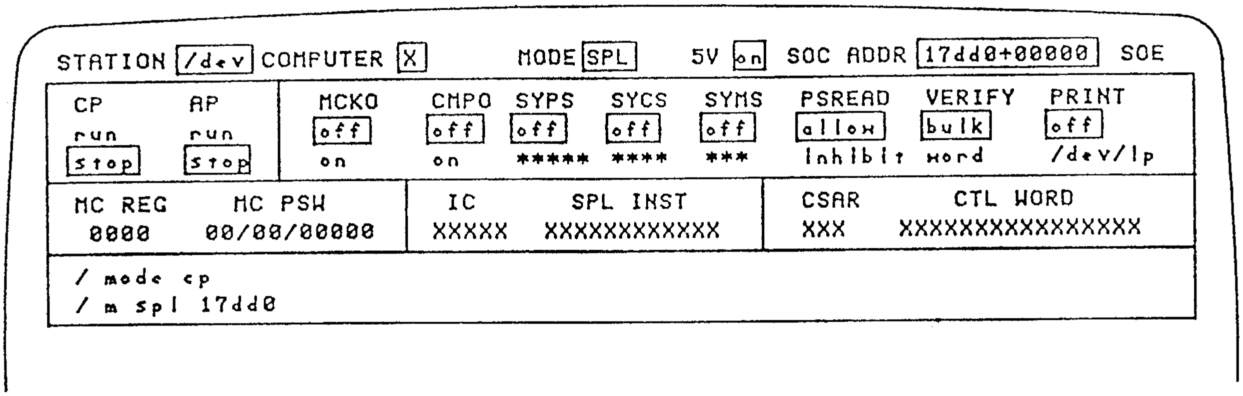
Example 1



```
STATION /dev  COMPUTER X        MODE CP     5V on  SOC ADDR 00000+00000  SOE
CP        AP      MCKO   CMPO  SYPS   SYCS  SYMS  PSREAD   VERIFY  PRINT
run       run     off    off   off    off   off   allow    bulk    off
stop      stop    on     on    *****  ****  ***   inhibit  word    /dev/lp
MC REG    MC PSW        IC     SPL INST       CSAR    CTL WORD
0000      00/00/00000   XXXXX  XXXXXXXXXXXX   XXX     XXXXXXXXXXXXXXXX
/
/ mode cp
```

In this example, the MODE command was executed to change from the SPL Mode to the CP mode as indicated by the Mode subfield.

Example 2



```
STATION [/dev] COMPUTER [X]      MODE [SPL]    5V [on]  SOC ADDR [17dd0+00000]  SOE

CP        AP       MCKO    CMPO   SYPS    SYCS   SYMS   PSREAD    VERIFY   PRINT
run       run      [off]   [off]  [off]   [off]  [off]  [allow]   [bulk]   [off]
[stop]    [stop]   on      on     *****   ****   ***    inhibit   word     /dev/lp

MC REG     MC PSW          IC       SPL INST           CSAR      CTL WORD
0000     00/00/00000       XXXXX    XXXXXXXXXXXX       XXX       XXXXXXXXXXXXXXXX

/ mode cp
/ m spl 17dd0
```

In this example, the MODE command was executed to set the mode to SPL as shown in the Mode subfield and to set the SOC address to 17dd0 as indicated by the SOC Addr subfield.

3.3.3.2.6 OVERRIDE

Command Description

This command will allow the user to override Machine Check errors and/or data/address compare errors. These overrides allow a MTS command to complete in spite of an error condition. For example, if part of a block of memory to be changed is protected, the user may override MCK errors in an attempt to change the non-protected addresses within the block. The override flags (i.e., MCKO and CMPO) will be set as defined by the overides parameter. The MCKO and CMPO subfield will be updated to show the present state of the override flags.

Command Syntax

o[verride] (nf)

where the nf parameter is defined as follows:

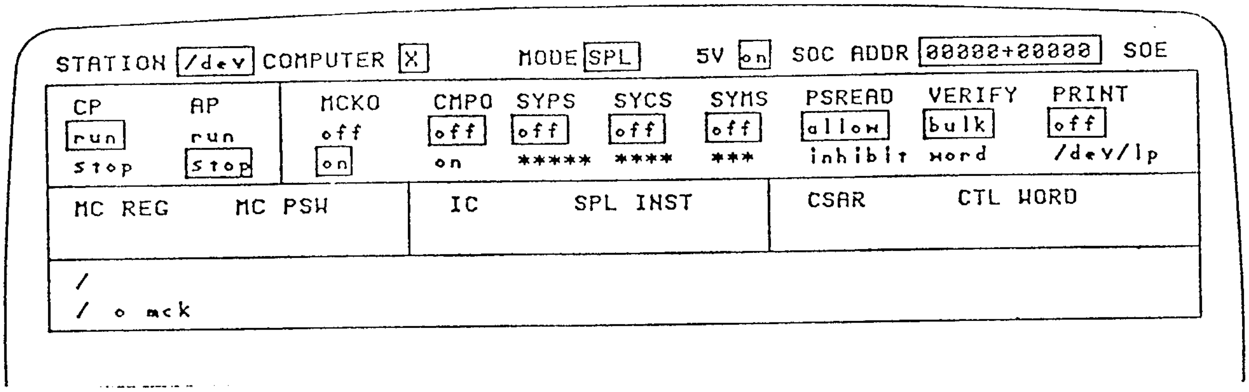
nf - off to turn overrides OFF

- mck to turn the MCK override ON

- cmp to turn CMP override ON

- both to turn both MCK and CMP overrides ON

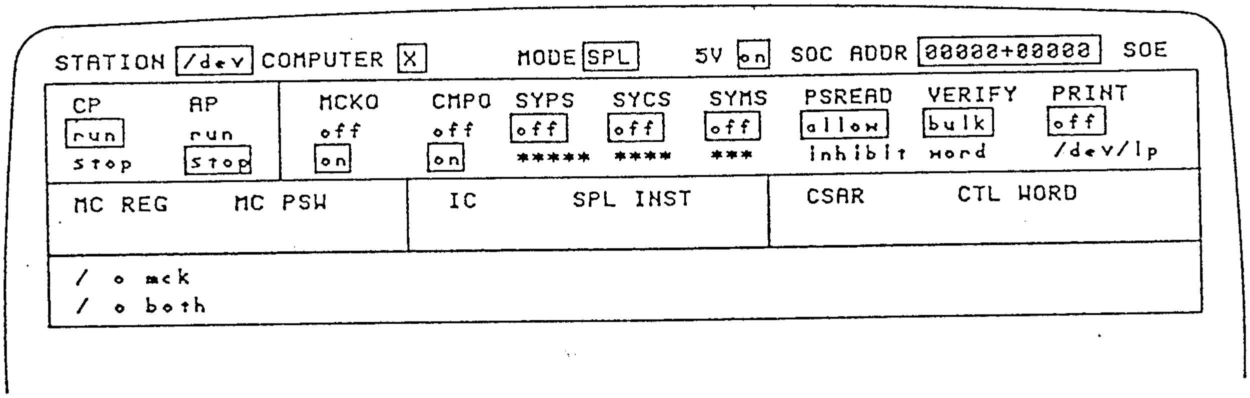
Example 1



```
STATION [/dev] COMPUTER [X]      MODE [SPL]     5V [on]   SOC ADDR [00002+00000]   SOE
 CP        AP      | MCKO    CMPO   SYPS    SYCS   SYMS   PSREAD    VERIFY   PRINT
[run]      run     | off    [off]  [off]   [off]  [off]  [allow]   [bulk]   [off]
 stop     [stop]   | [on]    on    *****   ****   ***    inhibit   word     /dev/lp
 MC REG    MC PSW        | IC        SPL INST        | CSAR        CTL WORD

 /
 / o mck
```

This example turns the MCK OVERRIDES ON as shown in the MCKO subfield.

Example 2



```
STATION [/dev] COMPUTER [X]      MODE [SPL]     5V [on]   SOC ADDR [00002+00000]   SOE
 CP        AP      | MCKO    CMPO   SYPS    SYCS   SYMS   PSREAD    VERIFY   PRINT
[run]      run     | off     off   [off]   [off]  [off]  [allow]   [bulk]   [off]
 stop     [stop]   | [on]   [on]   *****   ****   ***    inhibit   word     /dev/lp
 MC REG    MC PSW        | IC        SPL INST        | CSAR        CTL WORD

 / o mck
 / o both
```

This example turns both the MCK and CMP OVERRIDES ON as shown in the OVERRIDES subfield.

3.3.3.2.7 PRINT

Command Description

The PRINT command allows the operator to toggle the print switch on and off. When the print switch is on, all commands and function keys will be written to the defined print file, as will any data displayed in response to such commands. When the print switch is off, commands will be processed without writing to the defined print file. The print switch must be enabled in order to use the PRINT SCREEN command. The status of the PRINT command is displayed in the Print subfield of the MTS display. The MTS software will be initialized with the print switch set to "off".

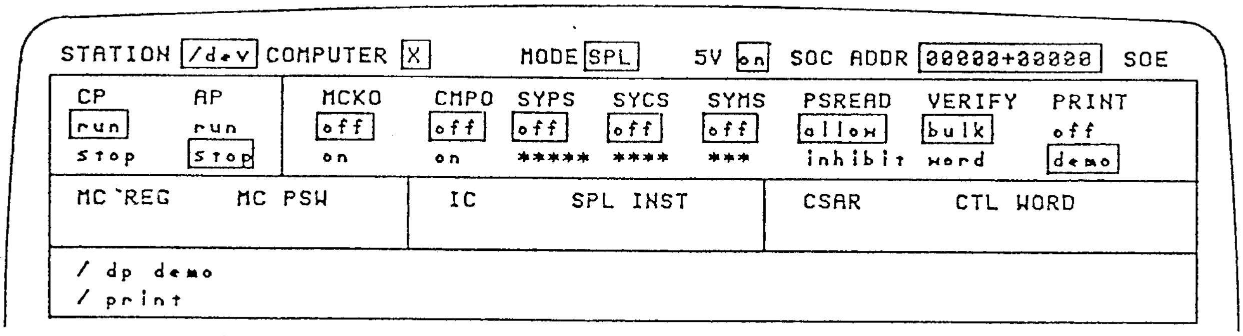
This command may also be activated using the numeric/function key <esc> 3.

Command Syntax

print

This command does not require any parameter definition.

Example



```
STATION /dev COMPUTER X        MODE SPL     5V on  SOC ADDR 00000+00000  SOE
 CP       AP      MCKO    CMPO   SYPS    SYCS   SYMS   PSREAD   VERIFY  PRINT
 run      run     off     off    off     off    off    allow    bulk    off
 stop     stop    on      on     *****   ****   ***    inhibit  word    demo
 MC REG     MC PSW     IC        SPL INST      CSAR        CTL WORD

 / dp demo
 / print
```

In this example Print has been enabled as indicated by the highlighted "demo" indication in Print subfield.

3.3.3.2.8 PRINT SCREEN

Command Description

This command will cause the current MTS display to be duplicated on the defined data storage device specified in the DEFINE PRINTER command. The FDD or RDD may be used to store the screen display. This duplicate screen will be printer formatted for printout at a later time. Some reformatting of the screen will be necessary since it is not possible to print video character attributes.

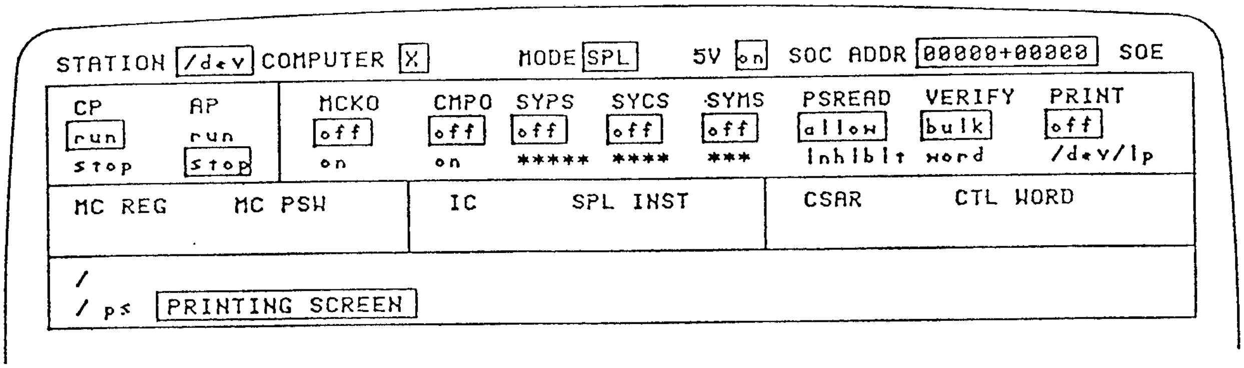
This command may also be activated using the numeric/function key <esc> F3.

Command Syntax

p[rint_]s[creen]

This command does not require any parameter definition.

Example



```
STATION [/dev] COMPUTER [X]      MODE [SPL]   5V [on]  SOC ADDR [00000+00000]  SOE
 CP        AP     | MCKO    CMPO   SYPS    SYCS   SYMS   PSREAD   VERIFY   PRINT
[run]      run    | [off]   [off]  [off]   [off]  [off]  [allow]  [bulk]   [off]
 stop     [stop]  |  on      on    *****   ****   ***    inhibit  word     /dev/lp
 MC REG      MC PSW     |  IC        SPL INST      |  CSAR       CTL WORD

/
/ ps  [PRINTING SCREEN]
```

This example shows the execution of the PRINT_SCREEN command. During execution, "PRINTING SCREEN" will be displayed on the present command line.

3.3.3.2.9 PSREAD

Command Description

When PSREAD is ALLOWed, the Register subfields of the MTS display will display data when the ASP is stopped. If CPSTEAL has also been enabled, the Register subfields will be dynamically updated once per second. When PSREAD is INHIBITed, no register data will be displayed. The PSREAD subfield will be updated to reflect the status of this switch. The default of the PSREAD mode is ALLOWed.

Command Syntax

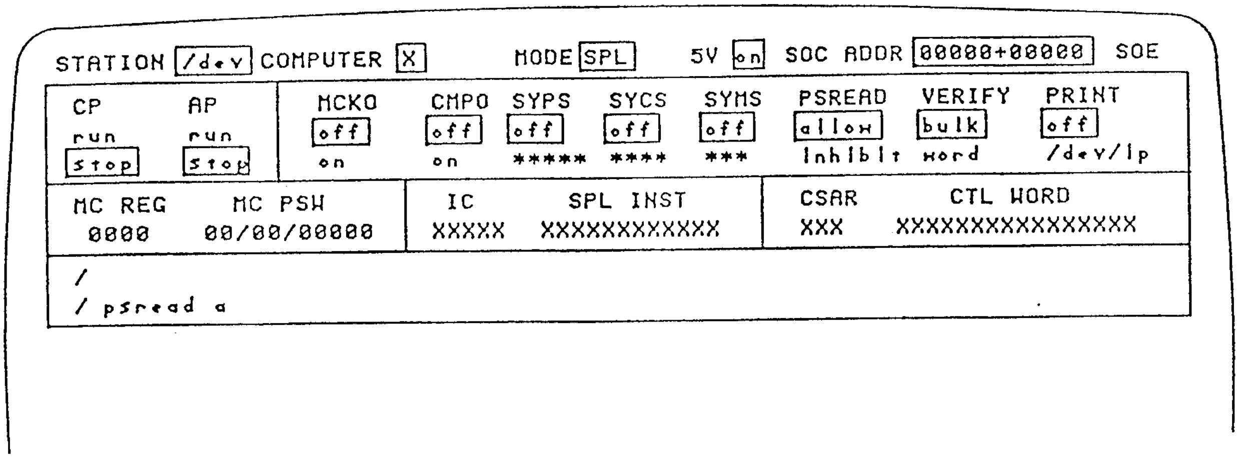
```
psread (nf)
```

Where the nf parameter is defined as follows:

a[llow] - PSREAD enabled

i[nhibited] - PSREAD disabled

Example



```
STATION /dev COMPUTER X      MODE SPL    5V on  SOC ADDR 00000+00000  SOE
CP     AP     MCKO  CMPO  SYPS   SYCS  SYMS  PSREAD   VERIFY  PRINT
run    run    off   off   off    off   off   allow    bulk    off
stop   stop   on    on    *****  ****  ***   inhibit  word    /dev/lp
MC REG   MC PSW        IC     SPL INST       CSAR   CTL WORD
0000     00/00/00000   XXXXX  XXXXXXXXXXXX   XXX    XXXXXXXXXXXXXXXX
/
/ psread a
```

This example enables the PSREAD as shown in the PSREAD Field. Since the CP is in the STOP mode, register information is displayed in the Register subfields.

3.3.3.2.10 RESET

Command Description

RESET will force a System Reset on the ASP. The CP Operational Microcode is forced to the specified reset address. If no reset address is specified (or if the terminal function key is used to invoke this command) the default address for the ASP will be used.

This command may also be activated using the numeric/function key <esc> 6.

Command Syntax

```
reset (ra)
```

where the parameters are defined as follows:

ra - reset address in control store

3.3.3.2.11 START

Command Description

If the current MODE is SPL or CP, the START command will start the ASP CP. If the current MODE is AP, the START command will start the ASP AP. If the applicable processor is already running, it will continue to RUN. The CP and AP subfields will be updated to indicate the change in processor state.

This command may also be activated using the numeric/function key <esc> F4.

Command Syntax

```
start
```

This command does not require any parameter definition.

3.3.3.2.12 START SOC

Command Description

The START SOC command works in conjunction with the MODE command to setup a Stop On Compare operation. The MODE command is used to specify an address which the ASP will stop on when found while fetching instructions. The SOC Address subfield will display the specified address. When the START SOC is activated, the CP and AP subfields will be updated to indicate the change in processor state and the "SOC" in the SOC Address subfield will blink until the next change in run rate. If the CS Stop-On Compare address is reached while the CP is running, the CP will enter CS-Step mode (see STEP).

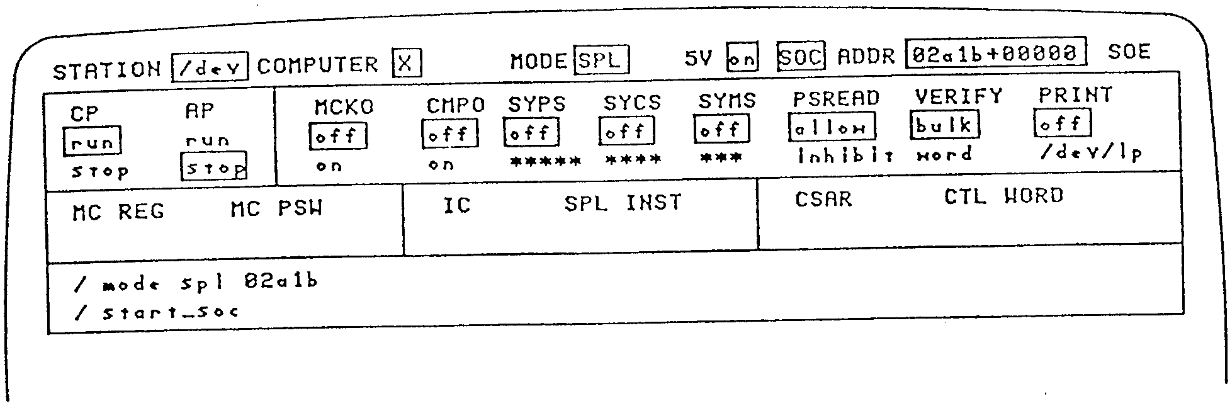
This command may also be activated using the numeric/function key <esc> F5.

Command Syntax

```
start_soc
```

This command does not require any parameter definition.

Example



```
STATION /dev COMPUTER X        MODE SPL     5V on  SOC ADDR 02a1b+00000  SOE
CP          AP       MCKO    CMPO  SYPS   SYCS  SYMS  PSREAD   VERIFY  PRINT
run         run      off     off   off    off   off   allow    bulk    off
stop        stop     on      on    *****  ****  ***   inhibit  word    /dev/lp
MC REG      MC PSW       IC        SPL INST           CSAR       CTL WORD

/ mode spl 02a1b
/ start_soc
```

This example will start the CP at the Start SOC rate. The SOC address is shown in the SOC Address subfield. Note that the SOC is also highlighted. This highlighting represents flashing.

3.3.3.2.13 START SOE

Command Description

The START SOE command will start the ASP computer designated by the MODE command. The CP and AP subfields will be updated to indicate the change in processor state and the SOE indicator will blink until the next change in run rate. The processor may subsequently stop if and when a Machine Check error occurs. If the current mode is AP, this command will be considered an error.

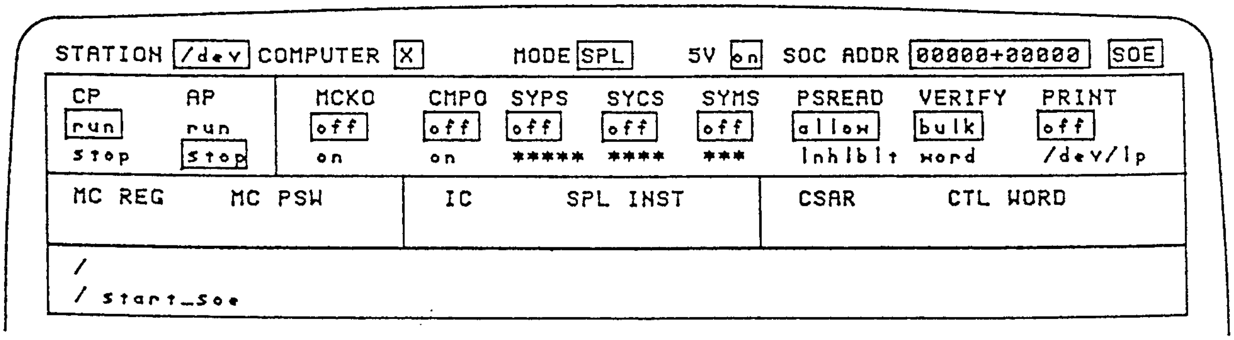
This command may also be activated using the numeric/function key <esc> F6.

Command Syntax

start_soe

This command does not require any parameter definition.

Example



```
STATION /dev COMPUTER X        MODE SPL     5V on  SOC ADDR 00000+00000  SOE
CP          AP       MCKO    CMPO  SYPS   SYCS  SYMS  PSREAD   VERIFY  PRINT
run         run      off     off   off    off   off   allow    bulk    off
stop        stop     on      on    *****  ****  ***   inhibit  word    /dev/lp
MC REG      MC PSW       IC        SPL INST           CSAR       CTL WORD

/
/ start_soe
```

In this example, the SOE indicator in the SOE subfield flashes while the SOE function is enabled.

3.3.3.2.14 STEP

Command Description

The STEP command will step the ASP computer designated by the MODE command one instruction. In SPL mode, PS is stepped, in CP mode, CS is stepped and in AP mode, MS is stepped. When CS is being stepped, the CP subfield will indicate "stopped". CS-Step mode is terminated by the next STOP, START, RESET, or MODE command.

This command may also be activated using the numeric/function key <esc> F7.

Command Syntax

step

This command does not require any parameter definition.

3.3.3.2.15 STOP

Command Description

The STOP command will stop the ASP computer specified in the MODE command. The CP and AP subfield will be updated to indicate the change in processor state.

This command may also be activated using the numeric/function key <esc> F7.

Command Syntax

stop

This command does not require any parameter definition.

3.3.3.2.16 SYNC

Command Description

The SYNC command is used to specify an address that the ASP will monitor and activate an interface line on the LPI interface. The user may monitor one of the following three SYNCs: PS, CS and MS SYNC. The SYNC subfield will display the address of the SYNC activated. Once the command is entered, the MTS software will monitor the sync operation. To display the SYNC status, the user must enter the DISPLAY SYNC STATUS command. If OFF is specified, the appropriate internal switch is set to indicate that SYNC is disabled; the SYNC subfield will be updated to indicate that the specified SYNC is disabled. The associated SYNC monitoring task will be terminated. Though the SYNC monitoring appears to be in the background mode, no other MTS commands may be executed while a SYNC is in progress.

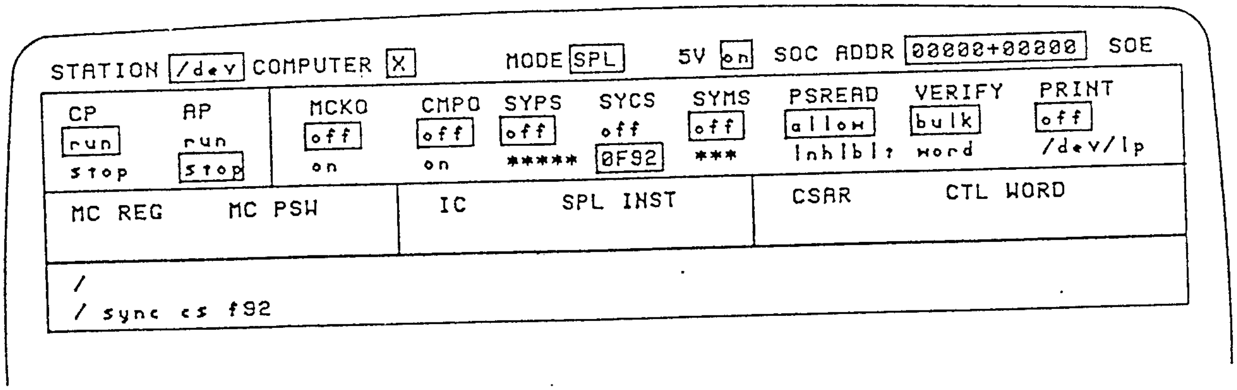
Command Syntax

```
     sync (st) (sa)
```

where the parameters are defined as follows:

```
     st - storage type (PS/CS/MS)
     sa - sync address
          OFF - disables the SYNC function
```

Example



```
STATION [/dev] COMPUTER [X]      MODE [SPL]    5V [on]  SOC ADDR [00000+00000]  SOE
 CP        AP      MCKO    CMPO   SYPS    SYCS   SYMS   PSREAD   VERIFY   PRINT
 [run]     run     [off]   [off]  [off]   off    [off]  [allow]  [bulk]   [off]
 stop      [stop]  on      on     *****   [0F92] ***    Inhibit  word     /dev/lp
 MC REG     MC PSW        IC        SPL INST         CSAR        CTL WORD

 /
 / sync cs f92
```

In this example the MTS software has been setup to Sync on Control Store address f92. To see SYNC status, the Display Sync Status command must be used.

3.3.3.2.17 VERIFY TYPE

Command Description

This command will set the internal verification type flag as specified. When subsequent processing involves modifying an ASP computer store, a verification type of WORD results in each standard unit of that store to be read after it is written for verification. A verification type of BULK results in the entire block of data being written, then the entire block being read back for verification. The Verify subfield will be updated to show the current verification type.

Command Syntax

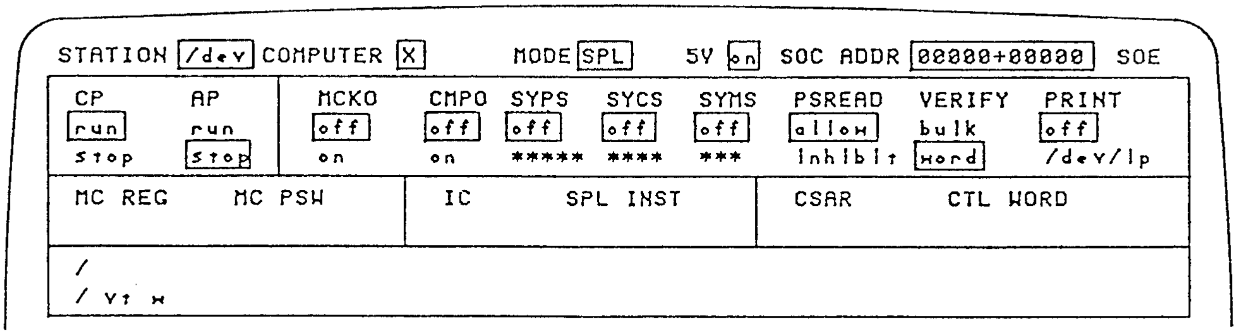
v[erify_]t[ype] (nf)

Where the nf parameter is defined as follows:

B[ULK] - Verify data on a block basis

W[ORD] - Verify data on a word basis

Example



```
STATION /dev COMPUTER X      MODE SPL   5V on  SOC ADDR 00000+00000  SOE

CP     AP     MCKO   CMPO   SYPS    SYCS   SYMS   PSREAD    VERIFY   PRINT
run    run    off    off    off     off    off    allow     bulk     off
stop   stop   on     on     *****   ****   ***    inhibit   word     /dev/lp

MC REG      MC PSW      IC        SPL INST       CSAR        CTL WORD

/
/ vt w
```

In this example, the MTS software has been changed from BULK to WORD as shown in the Verify Field of the MTS display.

3.3.3.3 ASP Memory/Register Commands

3.3.3.3.1 BASE

Command Description

The internal base address for all references to the specified ASP store shall be set to the specified value. The default value is zero.

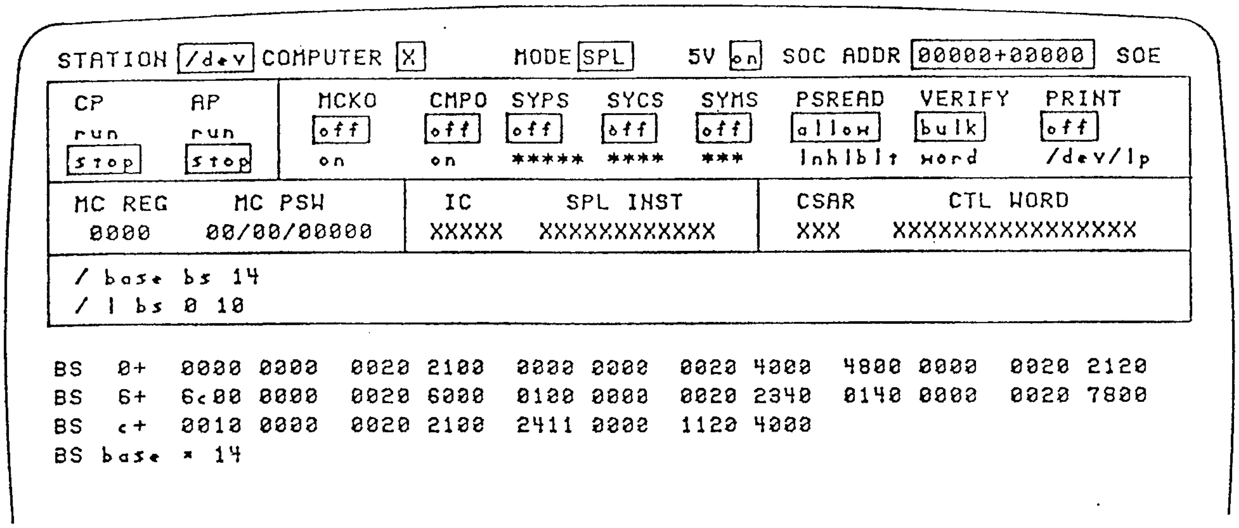
Command Syntax

b[ase] (st) [ba]

where the parameters are defined as follows:

st - storage type (PS, CS, BS, RS, MS, WO, W1 and R[EG])
ba - base address

Example



```
STATION /dev  COMPUTER X        MODE SPL    5V on  SOC ADDR 00000+00000  SOE

CP       AP       MCKO   CMPO   SYPS   SYCS   SYMS   PSREAD   VERIFY   PRINT
run      run      off    off    off    off    off    allow    bulk     off
stop     stop     on     on     *****  ****   ***    inhibit  word     /dev/lp

MC REG    MC PSW        IC      SPL INST        CSAR      CTL WORD
0000      00/00/00000   XXXXX   XXXXXXXXXXXX    XXX       XXXXXXXXXXXXXXXX

/ base bs 14
/ l bs 0 10

BS   0+   0000 0000   0020 2100   0000 0000   0020 4000   4800 0000   0020 2120
BS   6+   6c00 0000   0020 6000   0100 0000   0020 2340   0140 0000   0020 7800
BS   c+   0010 0000   0020 2100   2411 0000   1120 4000
BS base = 14
```

This example sets the base address of Bulk Store Memory to Hexidecimal 14.

3.3.3.3.2 CHANGE

Command Description

The CHANGE command will change the data in the designated memory locations to the specified values. The results will be displayed in the Display Data Field of the MTS display. First the original data in the memory location(s) will be displayed, followed by the new data that has just been changed. Verification will take place on a word by word basis or after the entire user specified block is written, as a function of the VERIFY TYPE command. If verification fails, an error message will be displayed in the Error Message Field of the MTS display.

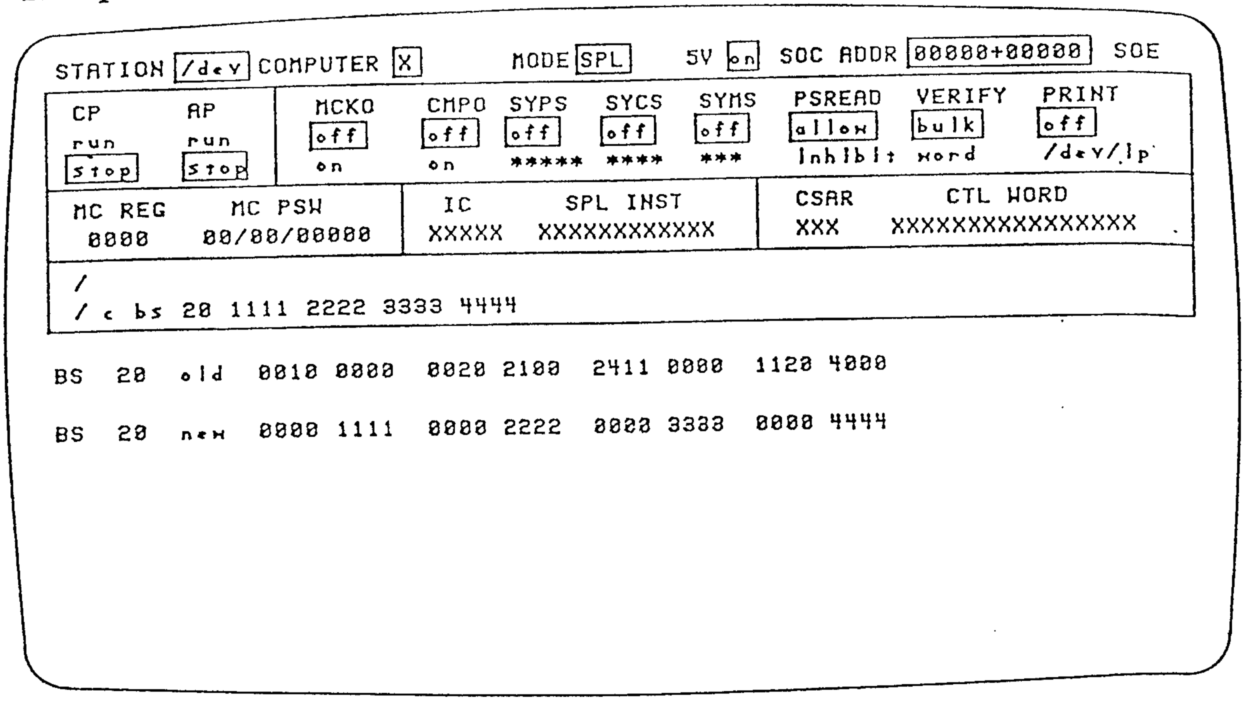
Command Syntax

```
c[hange] (st) (fa) (v1) [,v2,v3,...,vn]
```

where the parameters are defined as follows:

st - storage type (PS/CS/BS/RS/MS/WO/W1/R[EG])
fa - first address (or reg name GP, IC, MAR, CSAR,
vn - value to be stored/verified or "adr" (data=address pattern)

Example



```
STATION /dev COMPUTER X       MODE SPL    5V on  SOC ADDR 00000+00000  SOE
 CP        AP      MCKO    CMPO  SYPS   SYCS  SYMS  PSREAD   VERIFY  PRINT
 run       run     off     off   off    off   off   allow    bulk    off
 stop      stop    on      on    *****  ****  ***   inhibit  word    /dev/lp
 MC REG    MC PSW          IC      SPL INST         CSAR   CTL WORD
  0000     00/00/00000     XXXXX   XXXXXXXXXXXX     XXX    XXXXXXXXXXXXXXXX
 /
 / c bs 20 1111 2222 3333 4444

BS  20  old  0010 0000  0020 2100  2411 0000  1120 4000

BS  20  new  0000 1111  0000 2222  0000 3333  0000 4444
```

This example changes the first 4 locations of Bulk Store to 1111, 2222, 3333 and 4444.

3.3.3.3.3 CHANGE BLOCK

Command Description

The CHANGE BLOCK command will write the single value specified by the user into every address within the specified address range of the specified ASP store (note that the special value ADR assumes the numerical value of the address for each address written). Verification will take place as in CHANGE. This command will periodically indicate its progress during long block changes.

Command Syntax

c[hange_]b[lock] (st) (fa) (la) (vn)

where the parameters are defined as follows:

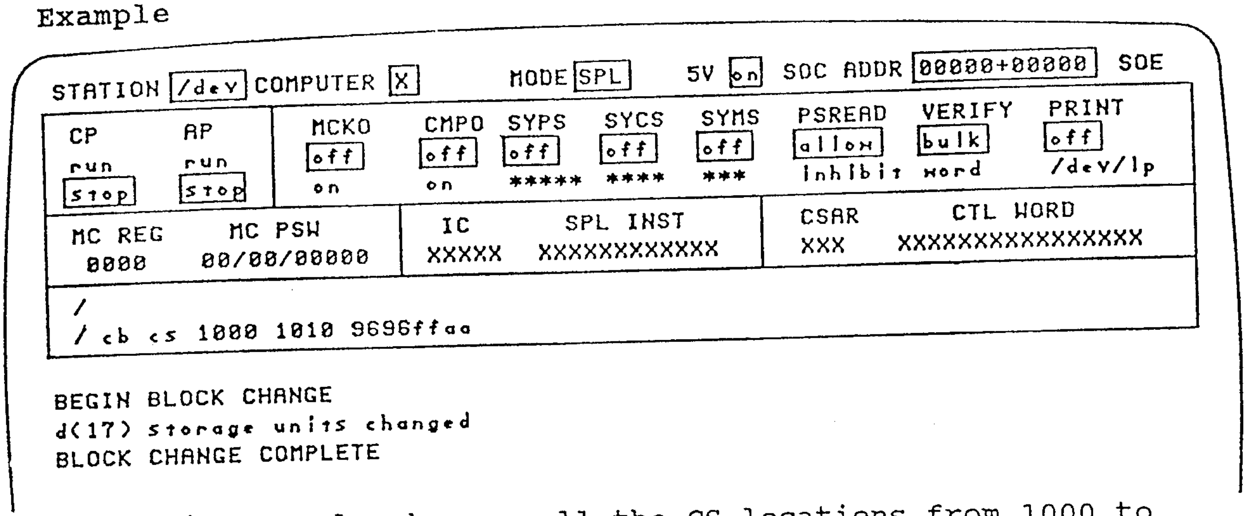
st - storage type (PS, CS, BS, RS, MS, WO, W1 and R[EG])

fa - first address (or reg name GP, IC, MAR, CSAR)

la - last address (or #count, or reg name)

vn - value to be stored/verified or "adr" (data=address pattern)

Example



```
STATION /dev COMPUTER X     MODE SPL    5V on  SOC ADDR 00000+00000  SOE
CP        AP       MCKO   CMPO  SYPS   SYCS  SYMS  PSREAD   VERIFY  PRINT
run       run      off    off   off    off   off   allow    bulk    off
stop      stop     on     on    *****  ****  ***   inhibit  word    /dev/lp
MC REG   MC PSW         IC      SPL INST        CSAR     CTL WORD
0000     00/00/00000    XXXXX   XXXXXXXXXXXX    XXX      XXXXXXXXXXXXXXXX
/
/ cb cs 1000 1010 9696ffaa

BEGIN BLOCK CHANGE
d(17) storage units changed
BLOCK CHANGE COMPLETE
```

This example changes all the CS locations from 1000 to 1010 to the Hexidecimal value 9696ffaa.

3.3.3.3.4 INSPECT

Command Description

The INSPECT command will fetch data from the specified ASP store, format into hexadecimal, and display it the in the Data Display Field. Each line displayed will have the storage type (i.e., PS, RS, BS, MS, CS, WO, W1, REG) in columns 1-3, the address (of the first data item displayed on the line) in columns 4-9, and 48 hex digits of data in columns 17-80 grouped with 1 space between each 4 hex digits and 2 spaces between each 8 hex digits. If a hardware register (CSAR, MAR, GP,IC, SCG, MI0, MI1, A00, A01) is to be inspected, only that single register needs to be entered as the first address (ie. no last address needed). The other register mnemonics (GP, IC, MAR) will be usable as synonyms for registers 20-3F, 48, and 60 (hex) respectively. When specified as a "last address", GP will be synonymous with 3f (hex). If a non-zero base is in effect (see BASE) for the specified store, the value of the base will also be displayed at the end of the inspected data display.

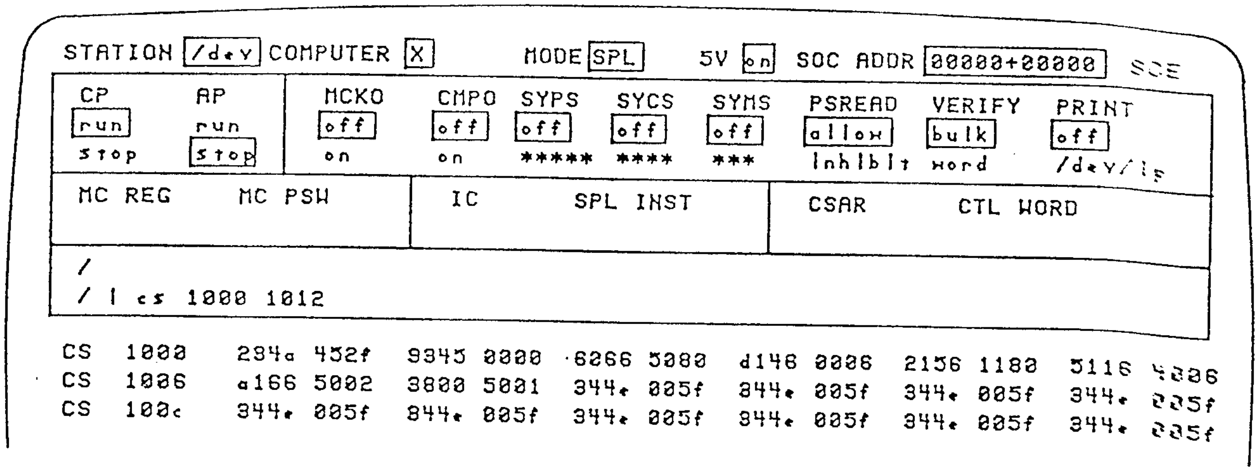
Command Syntax

```
i[nspect] (st) (fa) [la]
```

where the parameters are defined as follows:

st - storage type (PS, CS, BS, RS, MS, WO, W1 and R[EG])
fa - first address (or reg name GP, IC, MAR, CSAR, SCG, MI0, MI1, A00, A01)
la - last address (or #count, or reg name)

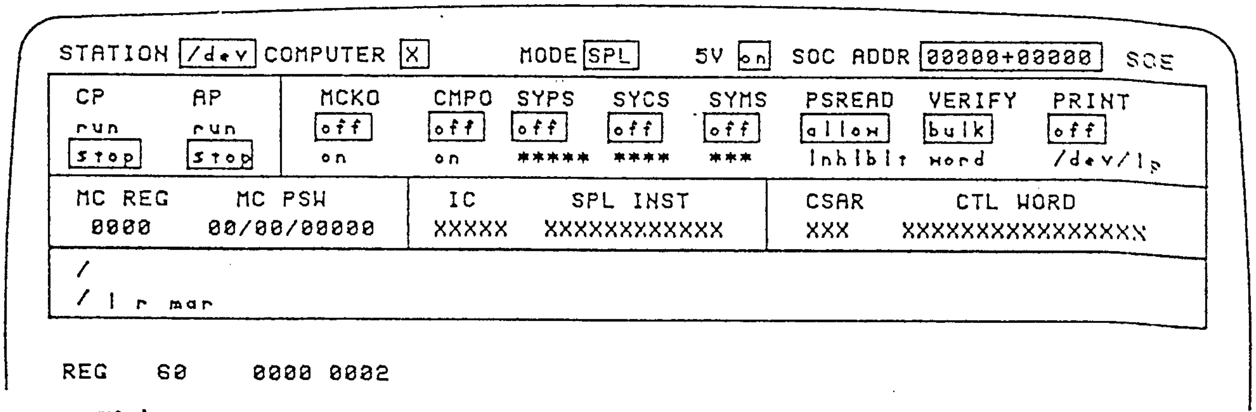
Example 1



```
STATION /dev COMPUTER X        MODE SPL    5V on  SOC ADDR 00000+00000  SCE
CP         AP       MCKO    CMPO   SYPS    SYCS   SYMS   PSREAD   VERIFY   PRINT
run        run      off     off    off     off    off    allow    bulk     off
stop       stop     on      on     *****   ****   ***    Inhibit  word     /dev/lp
MC REG     MC PSW            IC        SPL INST            CSAR       CTL WORD

/
/ | cs 1000 1012

CS   1000    234a 452f   9345 0000   6266 5080   d148 0006   2156 1180   5116 4006
CS   1006    a166 5002   3800 5001   344e 005f   844e 005f   344e 005f   344e 005f
CS   100c    344e 005f   844e 005f   344e 005f   844e 005f   344e 005f   344e 005f
```

In this example CS has been inspected from locations 1000 through 1012. The retrieved data is displayed in the Data Display Field.

Example 2



```
STATION /dev COMPUTER X        MODE SPL    5V on  SOC ADDR 00000+00000  SCE
CP         AP       MCKO    CMPO   SYPS    SYCS   SYMS   PSREAD   VERIFY   PRINT
run        run      off     off    off     off    off    allow    bulk     off
stop       stop     on      on     *****   ****   ***    Inhibit  word     /dev/lp
MC REG     MC PSW            IC        SPL INST            CSAR       CTL WORD
 0000      00/00/00000       XXXXX   XXXXXXXXXXXX          XXX   XXXXXXXXXXXXXXXX
/
/ | r mar

REG    60     0000 0002
```

This example retrieves the contents of the MAR register.

3.3.3.3.5 LOAD

Command Description

The LOAD command will retrieve data from a file that had been saved using the MTS SAVE command and restore the data to the specified ASP store in the memory range specified. The specified storage type, first address, and last address will be validated to ensure there is sufficient data on the file to load the specified storage area (if a smaller area is specified and there is more data on the file, excess data will be discarded and a warning message, with the amount of data discarded, will be written to the user terminal). Verification will take place on a word by word basis or on a block basis as a function of the VERIFY TYPE command. If verification fails, an error will be displayed in the Error Message Field. If the file does not contain a valid binary SAVE format, an error (FILE DOES NOT CONTAIN "SAVED" DATA) will be displayed in the Error Message Field and processing will be abandoned. This command will periodically indicate its progress during long loads.

Command Syntax

```
l[oad] (fn) [,st] [,fa] [,la]
```

where the parameters are defined as follows:

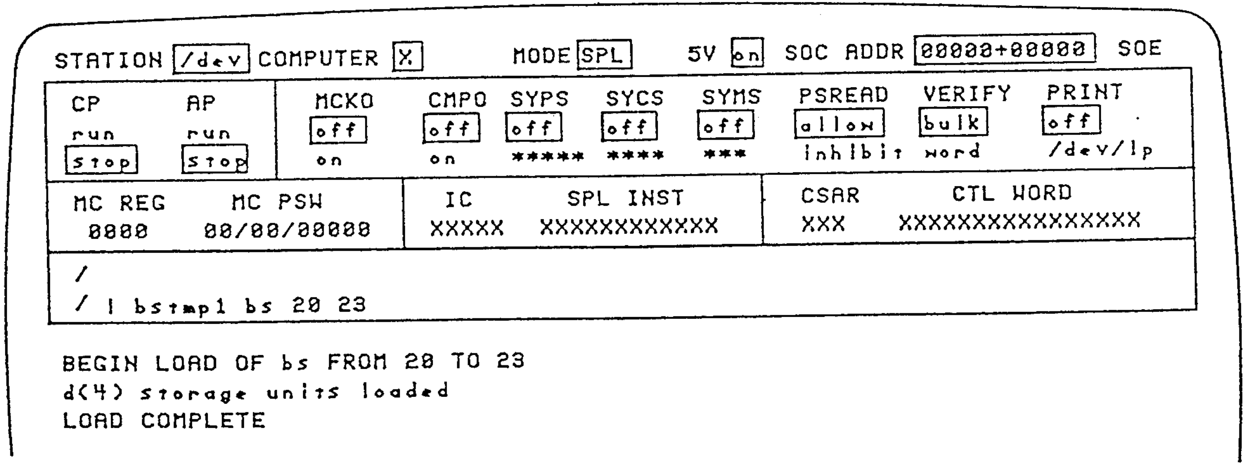
fn - file name

st - storage type (PS, CS, BS, RS, MS, WO, W1 and R[EG])

fa - first address (or reg name GP, IC, MAR, CSAR)

la - last address (or #count, or reg name)

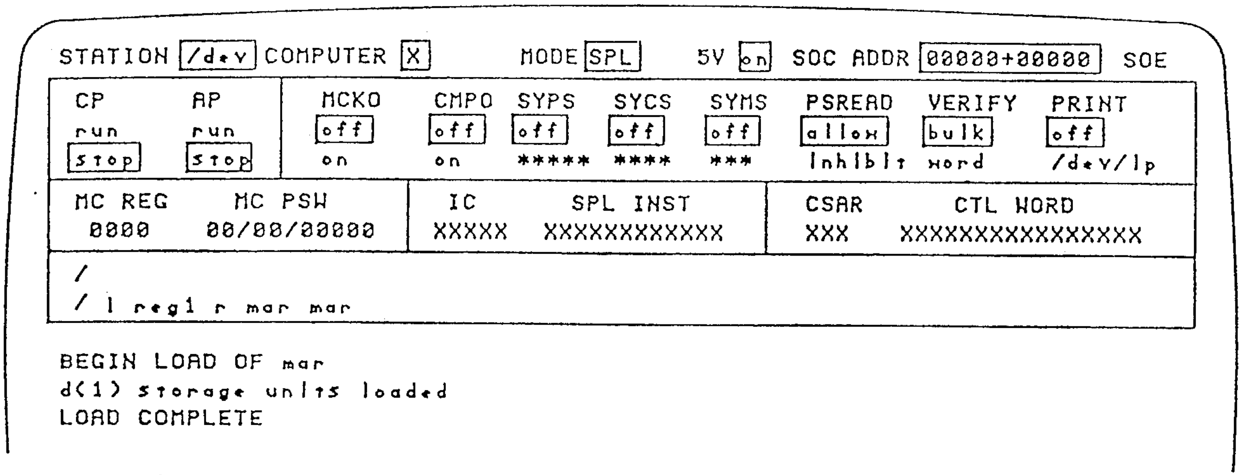
Example 1



```
STATION /dev COMPUTER X        MODE SPL     5V on  SOC ADDR 00000+00000  SOE
 CP       AP      MCKO    CMPO   SYPS    SYCS    SYMS   PSREAD   VERIFY   PRINT
 run      run     off     off    off     off     off    allow    bulk     off
 stop     stop    on      on     *****   ****    ***    inhibit  word     /dev/lp
 MC REG    MC PSW         IC      SPL INST          CSAR      CTL WORD
 0000     00/00/00000     XXXXX   XXXXXXXXXXXX      XXX   XXXXXXXXXXXXXXXX
 /
 / l bstmp1 bs 20 23

BEGIN LOAD OF bs FROM 20 TO 23
d(4) storage units loaded
LOAD COMPLETE
```

This example loads the contents of the bstmpl file into bulk store location 20 through 23.

Example 2



```
STATION /dev COMPUTER X        MODE SPL    5V on  SOC ADDR 00000+00000  SOE
 CP        AP       MCKO    CMPO  SYPS   SYCS   SYMS   PSREAD   VERIFY   PRINT
 run       run      off     off   off    off    off    allow    bulk     off
 stop      stop     on      on    *****  ****   ***    inhibit  word     /dev/lp
 MC REG     MC PSW         IC      SPL INST          CSAR       CTL WORD
  0000    00/00/00000      XXXXX   XXXXXXXXXXXX      XXX     XXXXXXXXXXXXXXXX
 /
 / l reg1 r mar mar

BEGIN LOAD OF mar
d(1) storage units loaded
LOAD COMPLETE
```

This example loads the contents of reg1 into the MAR register.

3.3.3.3.6 LOAD DISPLAY

Command Description

The LOAD DISPLAY command is used to complete the MODIFY DISPLAY operation, thus, it may only be used when a MODIFY DISPLAY command has been activated. After modifying the data in the Data Display Field, the user must then either depress the LOAD_DISPLAY function key or depress the HOME function key and enter the LOAD_DISPLAY command on the present command line. The displayed block of memory will be written back into the specified ASP store and verified as in CHANGE. Modification of the display will have no effect upon the corresponding ASP store unless the LOAD_DISPLAY command is processed while the modified data is still displayed.

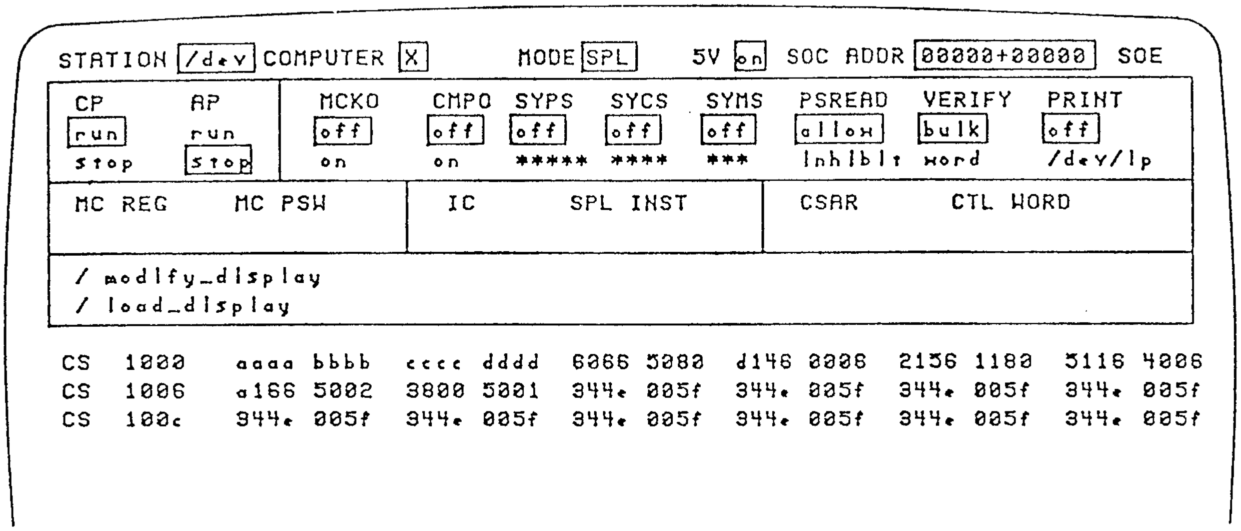
This command may also be activated using the numeric/function key <esc> 5.

Command Syntax

```
l[oad_]d[isplay]
```

This command does not require any parameter definition.

Example



```
STATION /dev COMPUTER X        MODE SPL     5V on  SOC ADDR 00000+00000  SOE

CP        AP      MCKO      CMPO  SYPS   SYCS   SYMS  PSREAD   VERIFY  PRINT
run       run     off       off   off    off    off   allow    bulk    off
stop      stop    on        on    *****  ****   ***   inhibit  word    /dev/lp

MC REG     MC PSW        IC         SPL INST         CSAR        CTL WORD

/ modify_display
/ load_display

CS   1000    aaaa bbbb  cccc dddd  6066 5080  d146 0006  2156 1180  5116 4006
CS   1006    a166 5002  3800 5001  344e 005f  344e 005f  344e 005f  344e 005f
CS   100c    344e 005f  344e 005f  344e 005f  344e 005f  344e 005f  344e 005f
```

In this example, the cursor has been returned to the present command line using the HOME numeric/function key after performing a MODIFY_DISPLAY operation. The LOAD_DISPLAY command must be executed to make the changes to the ASP store.

3.3.3.3.7 MODIFY DISPLAY

Command Description

The MODIFY DISPLAY command will enable the operator to edit the data displayed in the Data Display Field. This command will be valid only if the Data Display Field of the MTS display contains data from a previous INSPECT command. The user will be permitted, when such data is displayed, to move the cursor into the displayed data portion of the Data Display Field written by INSPECT and to modify the data found there. This is accomplished by pressing <esc> j. If during such modification, the user attempts to modify a field other than the data hexadecimal digits or if he attempts to enter a non-hexadecimal digit in a valid field, the character entered will not be displayed and the keystroke will be ignored. Modified digits will be displayed at a higher brightness than the original unmodified digits. If the MODIFY DISPLAY is executed without first performing an INSPECT, an error (inspected storage not currently displayed) will be displayed in the Error Message Field of the MTS display.

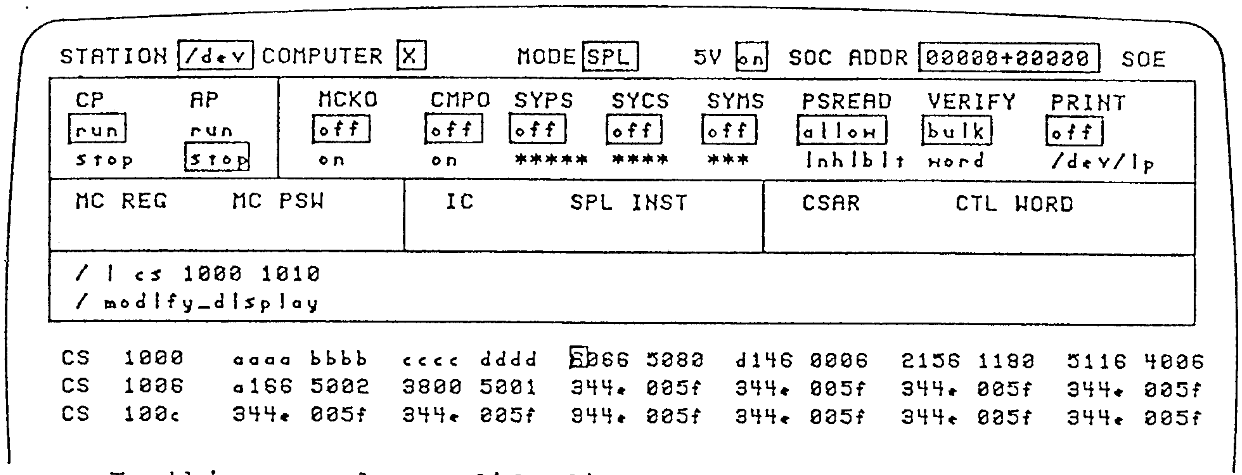
This command may also be activated with the numeric/function key <esc> j.

Command Syntax

modify_display

This command does not require any parameter definition.

Example



```
STATION /dev COMPUTER X        MODE SPL     5V on  SOC ADDR 00000+00000  SOE

 CP        AP      MCKO     CMPO  SYPS   SYCS   SYMS   PSREAD   VERIFY   PRINT
 run       run     off      off   off    off    off    allow    bulk     off
 stop      stop    on       on    *****  ****   ***    inhibit  word     /dev/lp

 MC REG     MC PSW       IC        SPL INST          CSAR         CTL WORD

 / l cs 1000 1010
 / modify_display

CS   1000     aaaa bbbb   cccc dddd   5066 5080   d146 0006   2156 1180   5116 4006
CS   1006     a166 5002   3800 5001   344e 005f   344e 005f   344e 005f   344e 005f
CS   100c     344e 005f   344e 005f   344e 005f   344e 005f   344e 005f   344e 005f
```

In this example, modify_display has been entered on the present command line which moved the cursor into the Data

Display Field. As shown above, four values have been changed to aaaa, bbbb, cccc and dddd. The cursor will remain in the Data Display Field until either the the LOAD-DISPLAY or the HOME numeric/function key has been pressed.

3.3.3.3.8 SAVE

Command Description

The SAVE command allows the user to save an image of the specified ASP store. When invoked, the data within the specified memory range will be fetched from the appropriate ASP store, blocked into binary records, and written to the specified file. This command will periodically indicate its progress during longer SAVEs.

Command Syntax

```
s[ave] (fn) (st) (fa) (la)
```

where the parameters are defined as follows:

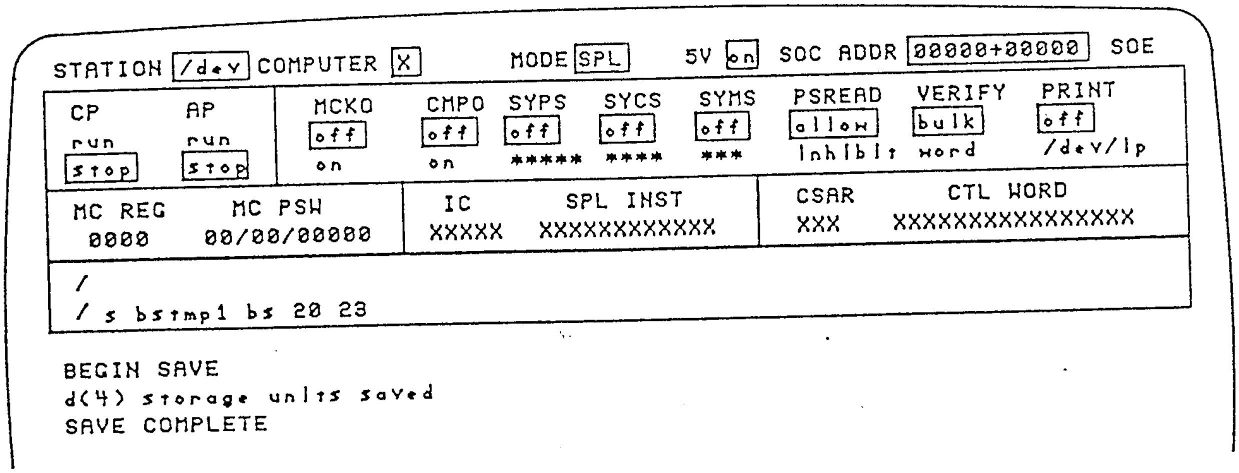
fn - file name

st - storage type (PS, CS, BS, RS, MS, W0, W1 and R[EG])

fa - first address (or reg name GP, IC, MAR, CSAR, SCG, MI0, MI1, A00, A01)

la - last address (or #count, or reg name)

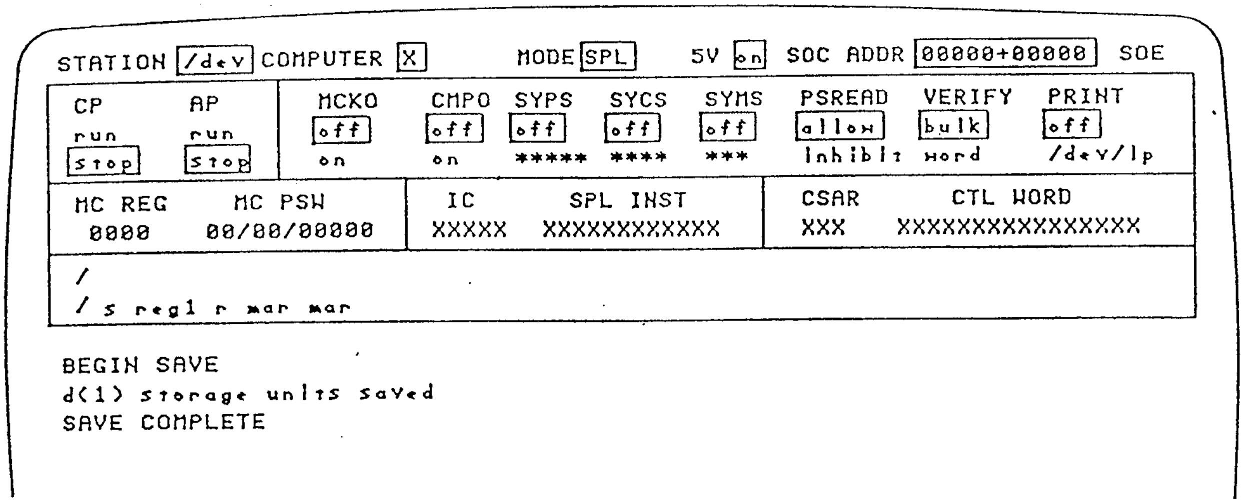
Example 1



```
STATION [/dev] COMPUTER [X]      MODE [SPL]    5V [on]  SOC ADDR [00000+00000]  SOE

CP        AP        MCKO    CMPO    SYPS    SYCS    SYMS    PSREAD    VERIFY   PRINT
run       run       [off]   [off]   [off]   [off]   [off]   [allow]   [bulk]   [off]
[stop]    [stop]    on      on      *****   ****    ***     inhibit   word     /dev/lp

MC REG    MC PSW          IC      SPL INST          CSAR    CTL WORD
0000      00/00/00000     XXXXX   XXXXXXXXXXXX      XXX     XXXXXXXXXXXXXXXX

/
/ s bstmp1 bs 20 23

BEGIN SAVE
d(4) storage units saved
SAVE COMPLETE
```

This example saves the contents of bulk store from locations 20 through 23 to a file named bstmp1.

Example 2



```
STATION [/dev] COMPUTER [X]      MODE [SPL]   5V [on]  SOC ADDR [00000+00000]  SOE

 CP        AP       MCKO    CMPO  SYPS   SYCS  SYMS  PSREAD   VERIFY  PRINT
 run       run      [off]   [off] [off]  [off] [off] [allow]  [bulk]  [off]
[stop]    [stop]     on      on   *****  ****  ***   inhibit  word    /dev/lp

 MC REG    MC PSW        IC      SPL INST        CSAR      CTL WORD
  0000   00/00/00000     XXXXX   XXXXXXXXXXXX    XXX   XXXXXXXXXXXXXXXX

 /
 / s reg1 r mar mar

BEGIN SAVE
d(1) storage units saved
SAVE COMPLETE
```

This example saves the contents of the MAR register to a file named reg1.

3.3.3.3.9 SEARCH

Command Description

The SEARCH command will fetch data from the appropriate ASP store beginning at the specified first address. Each unit fetched will be ANDed with the mask and compared with the value specified (also ANDed with the mask). When a match occurs, it will be displayed in the Data Display Field. The output line shall contain the storage type in columns 1-3, the address in columns 4-9, and the data value at that address (4, 8, or 16 hexadecimal digits depending on storage type) in columns 17-36, spaced as in INSPECT. The search will then continue until the number of displayed matches equals the specified count or the last address specified has been fetched and compared. If no match has been found after the data is fetched from the last address specified, "NOT FOUND" will be displayed and the search terminated. The default count will be 1 and the default mask shall be FFFFFFFFFFFFFFFF. This command will periodically indicate its progress during long searches.

Command Syntax

    search (st), (fa), (la), (vn) [,mk] [,ct]

where the parameters are defined as follows:

st - storage type (PS, CS, BS, RS, MS, W0, W1 and R[EG])

fa - first address (or reg name GP, IC, MAR, CSAR, SCG, MI0, MI1, A00, A01)

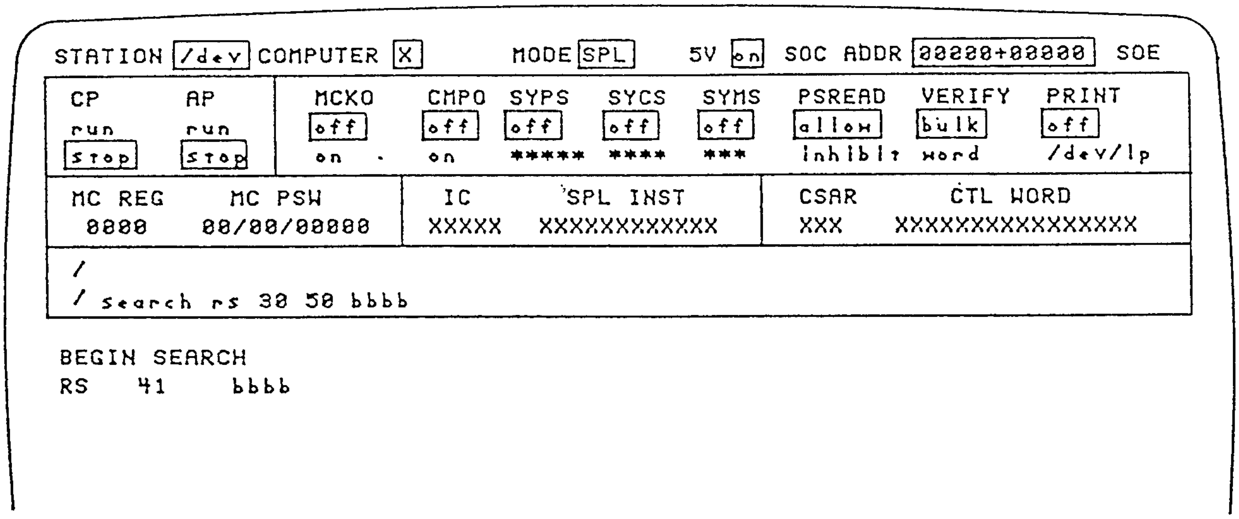
la - last address (or #count, or reg name)

vn - value to be searched for

mk - mask value

ct - count

Example



```
STATION /dev COMPUTER X        MODE SPL    5V on  SOC ADDR 00000+00000  SOE
 CP        AP      MCKO     CMPO  SYPS   SYCS  SYMS  PSREAD  VERIFY  PRINT
 run       run     off      off   off    off   off   allow   bulk    off
 stop      stop    on       on    *****  ****  ***   inhibit word    /dev/lp
 MC REG     MC PSW         IC      SPL INST          CSAR     CTL WORD
  0000     00/00/00000     XXXXX   XXXXXXXXXXXX      XXX      XXXXXXXXXXXXXXXX
 /
 / search rs 30 50 bbbb

BEGIN SEARCH
RS    41     bbbb
```

This example searches Reserve Store for the pattern bbbb from location 30 to 50. A match has been found at location 41.

3.3.3.3.10 +SCROLL

Command Description

This command will function only if the previous command was INSPECT. The INSPECT command will be repeated with its first address and last address adjusted to display the next block (of the same size and same storage type) of data. No data will be displayed if the previous command was not INSPECT.

This command may also be activated using the numeric/function key <esc> F1.

Command Syntax

```
+[scroll]
```

This command does not require any parameter definition.

3.3.3.3.11 -SCROLL

Command Description

This command will function only if the previous command was INSPECT. The INSPECT command will be repeated with its first address and last address adjusted to display the previous block (of the same size and same storage type) of data. No data will be displayed if the previous command was not INSPECT.

This command may also be activated using the numeric/function key <esc> F2.

Command Syntax

```
-[scroll]
```

This command does not require any parameter definition.

3.3.3.3.12 VERIFY

Command Description

VERIFY processing will be similar to LOAD processing except the specified ASP store will not be modified but rather compares ASP store to a file saved using the SAVE command. Each address which does not contain the specified value will be displayed in the Data Display Field. Each line shall contain the storage type in columns 1-3, the address in columns 4-9, the word "Expected--" in columns 17-26, the word "Actual----" in columns 50-59, and the hexadecimal expected and actual values (4, 8, or 16 hex digits, depending on the storage type) in columns 28-47 and 61-80 respectively (spaced as in INSPECT). This display will be provided instead of displaying an error in the Error Message Field. If the verify is completely successful, "VERIFY GOOD" will be displayed. This command will periodically indicate its progress during long verifies.

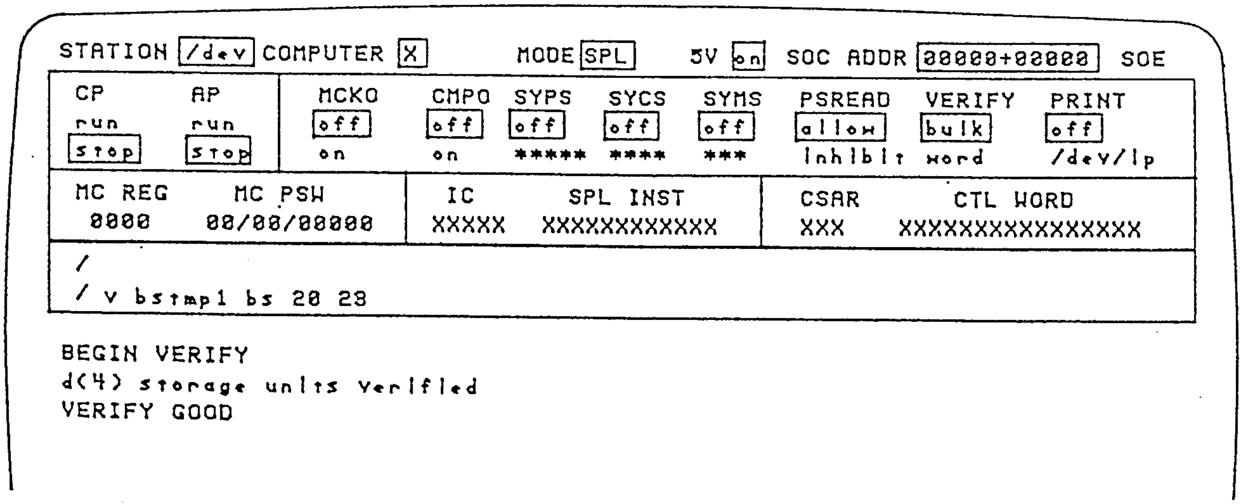
Command Syntax

```
v[erify] (fn) [st] [fa] [la]
```

where the parameters are defined as follows:

fn - file name
st - storage type (PS, CS, BS, RS, MS, WO, W1 and R[EG])
fa - first address (or reg name GP, IC, MAR, CSAR, SCG,
     MI0, MI1, AO0, AO1)
la - last address (or #count, or reg name)

Example



```
STATION [/dev] COMPUTER [X]     MODE [SPL]   5V [on]  SOC ADDR [00000+00000]  SOE

 CP       AP       MCKO    CMPO   SYPS    SYCS   SYMS   PSREAD   VERIFY  PRINT
 run      run      [off]   [off]  [off]   [off]  [off]  [allow]  [bulk]  [off]
 [stop]   [stop]    on      on    *****   ****   ***    inhibit  word    /dev/lp

 MC REG     MC PSW        IC      SPL INST        CSAR      CTL WORD
  0000    00/00/00000     XXXXX   XXXXXXXXXXXX    XXX    XXXXXXXXXXXXXXXX

 /
 / v bstmp1 bs 20 23

BEGIN VERIFY
d(4) storage units verified
VERIFY GOOD
```

In this example, the VERIFY command is used to compare the contents of Bulk Store locations 20 through 23 with the contents stored in the file bstmp1. The verify operation was successful.

3.3.3.3.13 VERIFY BLOCK

Command Description

The VERIFY BLOCK command will fetch data from each address within the specified address range of the specified ASP store and compare it with the single value specified by the user (note that the special value ADR assumes the numerical value of the address for each address read). The results of this verification will be displayed in the Data Display Field as described in the VERIFY command.

Command Syntax

```
v[erify_]b[lock] (st), (fa), (la), (vn)
```

where the parameters are defined as follows:

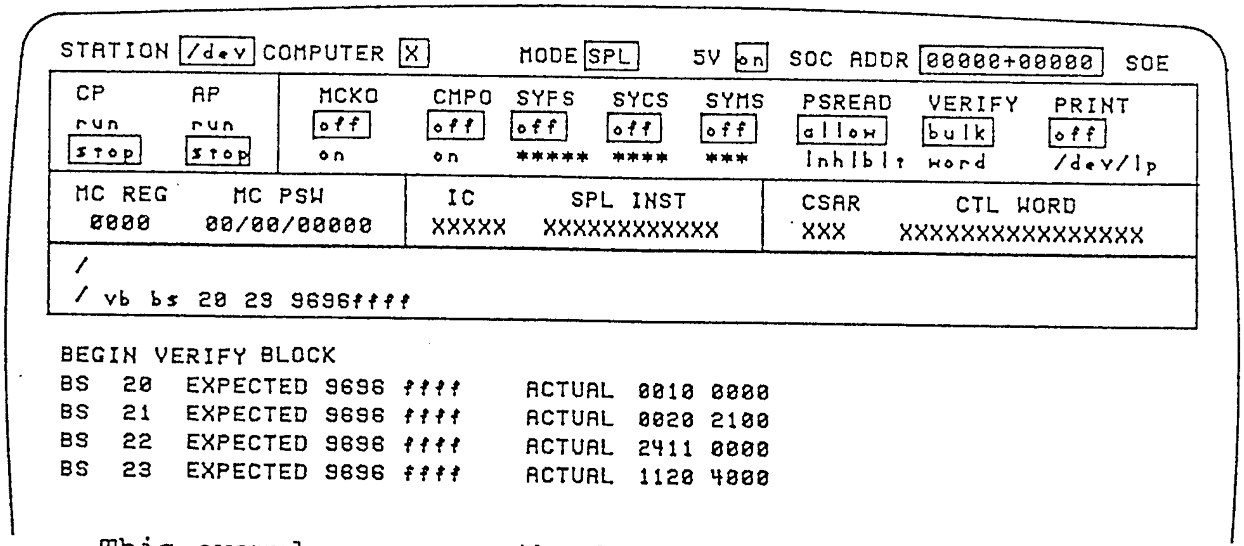
st - storage type (PS, CS, BS, RS, MS, WO, W1 and R[EG])

fa - first address (or reg name GP, IC, MAR, CSAR, SCG, MI0, MI1, AO0, AO1

la - last address (or #count, or reg name)

vn - value to be verified or "adr" (data=address pattern)

Example



```
STATION [/dev] COMPUTER [X]      MODE [SPL]    5V [on]  SOC ADDR [00000+00000]  SOE

CP       AP       MCKO    CMPO   SYFS    SYCS   SYMS   PSREAD    VERIFY   PRINT
run      run      [off]   [off]  [off]   [off]  [off]  [allow]   [bulk]   [off]
[stop]   [stop]   on      on     *****   ****   ***    inhibit   word     /dev/lp

MC REG    MC PSW        IC      SPL INST        CSAR     CTL WORD
0000      00/00/00000   XXXXX   XXXXXXXXXXXX    XXX      XXXXXXXXXXXXXXX

/
/ vb bs 20 23 9696ffff

BEGIN VERIFY BLOCK
BS   20   EXPECTED 9696 ffff     ACTUAL  0010 0000
BS   21   EXPECTED 9696 ffff     ACTUAL  0020 2100
BS   22   EXPECTED 9896 ffff     ACTUAL  2411 0000
BS   23   EXPECTED 9896 ffff     ACTUAL  1120 4000
```

This example compares the data located in Bulk Store locations 20 thorugh 23 with the specified constant 9696fff. All compares failed.

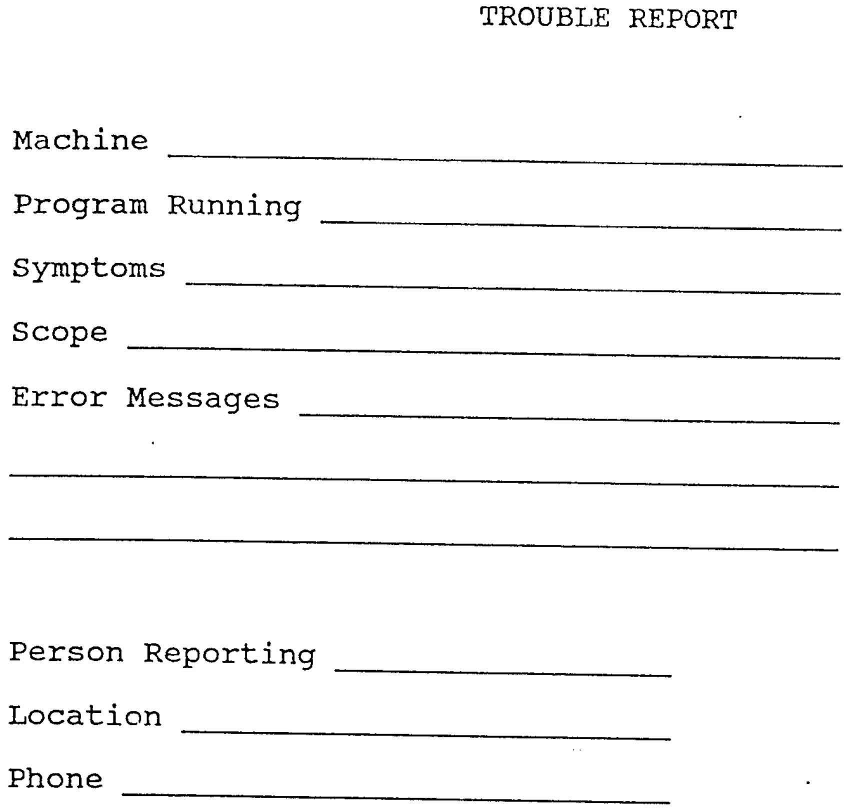
3.4 Preventative Maintenance

The TU is a very reliable work station. The components implemented in the design of the TU were selected to reduce the amount of preventative maintenance required. The only component that requires any preventative maintenance is the TU display. The TU display is an electroluminent flat panel display which has an amber polarized filter on the front to prevent reflection and glare. It is very important not to scratch the viewing area. When cleaning is required, clean with non-abrasive mild glass cleaner.

Other than the periodic cleaning of the display, there is no other preventative maintenance required.

3.4.1 In Case of a Problem

In the event of a problem, it is very important to record all conditions leading up to the event. Figure 3-11 shows a sample Trouble Report that can be used to record and keep track of system problems. Appendix E contains a list of debug commands which can be executed to further locate a problem area.



TROUBLE REPORT

Machine ______________________________

Program Running ______________________

Symptoms _____________________________

Scope ________________________________

Error Messages _______________________

____

____

Person Reporting _______________

Location _______________________

Phone __________________________

Figure 3-11. TU Sample Trouble Report

Appendix A

PAWSWORK Operator Interface

Command Quick Reference

PAWSWORK Operator Interface Commands

Command Syntax	Page
bye	43
catlist [s[ou]rce=path1] [place=path2]	37
cg[lobal] [p1] [p2] [p3] ... [pn]	46
config[ure] [std=stdconf] [op=operation]	44
copy s[ou]rce=path1 [dest[ination]=path2] [form[at]=fchar]	38
ex[amine] [file=path1] [form[at]=fchar]	39
gen[data] dest[ination]=path2 [form[at]=fchar]	40
help cmd=PAWSWORK command	44
lg[lobal] [p1] [p2] [p3] ... [pn]	46
mod[ify] file=path1 [dest[ination]==path2]	40
mts	49
permit file=path1 access=access code	41
pr[int] file=path1 [place= path2] [form[at]=fchar]	42
purge file=path1	42
resume	45
sg[lobal] [parameter keyword 1=parameter value 1] [parameter keyword 2=parameter value 2]... [parameter keyword n=parameter value n]	47

The parameters shown in brackets are optional.

Appendix B

MTS Command Quick Reference

MTS Commands

The user may enter a command in the form of a text line terminated by a carriage return in the User Entry Field. The user may also enter commands which have no parameters by pressing a numeric or function key on the TU keyboard.

MTS Support Commands

Numeric/ Function Key	Command Syntax	Page
	COMMENT (dt)	58
	ECP41	59
ESC 8	END	59
	HELP	60
ESC 2	HOME	61
	INCLUDE (fn)	62
ESC 0	INTERRUPT	63
ESC 7	NEXT	63
ESC 1	R[EPEAT] [ct]	64
	R[EPEAT_]H[ARDWARE] [R]	65
	SCREEN [nf]	66
ESC 9	SUSPEND	67
	T[EST_]H[ARDWARE]	68
	WAIT [ct]	68
ESC 2	WATCH	69

ASP Control and Status Commands

Numeric/ Function Key	Command Syntax	Page
ESC 4	CPSTEAL	70
	DEFINE PRINTER	71
	D[ISPLAY_]S[YNC_]S[TATUS]	71
	FORCE (CW/AECW) (in)	72
	M[ODE] (SPL/CP/AP) [ca]	73
	O[VERRIDE] (OFF/MCK/CMP/BOTH)	74
ESC 3	PRINT	75
ESC F3	P[RINT_]S[CREEN]	76
	PSREAD [A[LLOW]/I[NHIBIT]]	77
ESC 6	RESET [ra]	78
ESC F4	START	79
ESC F5	START_SOC	79
ESC F6	START_SOE	80
ESC F7	STEP	81
ESC F7	STOP	81
	SYNC (st) (sa/OFF)	81
	V[ERIFY_]T[YPE] (B[ULK]/W[ORD])	82

ASP Memory and Register Commands

Numeric/ Function Key	Command Syntax	Page
	B[ASE] (st) [ba]	83
	C[HANGE] (st) (fa) (v1) [v2 v3 ... vn]	84
	C[HANGE_]B[LOCK] (st) (fa) (la) (vn)	85
	I[NSPECT] (st) (fa) [la]	86
	L[OAD] (fn) [st] [fa] [la]	87
ESC 5	L[OAD_]D[ISPLAY]	89
	MODIFY_DISPLAY (dt)	90
	S[AVE] (fn) (st) (fa) (la)	91
	SEARCH (st) (fa) (la) (vn) [mk] [ct]	93
ESC F1	+[SCROLL]	95
ESC F2	-[SCROLL]	95
	V[ERIFY] (fn) [st] [fa] [la]	96
	V[ERIFY_]B[LOCK] (st) (fa) (la) (vn)	97

where the parameters are defined as follows:

st - storage type PS, CS, BS, RS, MS, W0, W1 and R[EG]
fa - first address (or reg name GP, IC, MAR, CSAR, SCG, MI0, MI1, AO0, AO1
la - last address (or #count, or reg name)
vn - value to be stored/verified or "ADR" (data=address pattern)
fn - file name
mk - mask value
sa - sync address
in - an instruction word
ca - compare address for stop-on-compare
ba - base address
dn - device name
ct - count
ra - reset address in control store
pa - program store address
qc - quick stop class
dt - data
nf - on or off

Appendix C

MTS Error Message Summary

MTS Error Message Summary

System Error

Unimplemented command

Token inconsistency

No maintenance channels configured

Configuration inconsistency

Cannot open maintenance channel

Too many parameters specified

Too few parameters specified

Invalid key depression

Invalid command

Unacceptable file/device name

File name too long; max char's =

Invalid mode

Invalid ASP storage type

Invalid address

Address exceeds available storage

Invalid count

First address exceeds last

Count exceeds available storage

Previous command not repeatable

Previous command not valid inspect

Try \"help\" - No help available for

Invalid override class

Invalid psread class

MTS Error Message Summary (Continued)

Invalid verify type

Invalid mode

Value too large for specified store

Invalid value

Stop-On-Error only valid in SPL mode

Range may not include a hardware register

Base may not be set to hardware register

Inspected storage not currently displayed

File not in \"save\" format

Specify both FWA/LWA or neither - base not zero

Rewrite not confirmed for

Range may not include both Local and Primary register

Maintenance Panel Interface - No Power!

ASP Computer - No Power!

Power restored to

Warning - Change does not verify invalid sync storage type

Sync type not compatible with mode

Invalid quick_stop action

Unknown error code

Note: When a message listed above is shown in the Error Message Field it is preceded by the computer type and the time of day.

Appendix D

Quick Reference

for

Modem Commands

Command Description

Command		Description (* denotes the factory setting)
AT		Attention Code (Command Sequence Prefix)
A/		Repeat Preceding Command Sequence
A		Answer Call Immediately
B0		CCITT Answer Tone
B1	*	BELL Answer Tone
Dn		Dial Telephone Number (n)
		Additional Dial Modifier Commands:
		T = Touch Tone Mode
	*	P = Pulse Dial Mode
		R = Reverse To Answer Mode After Dialing
		S = Dial Stored Number
		W = Wait For Dialtone
		, = Pause (2 Seconds)
		; = Return To Command State After Dialing
		@ = Wait For Quiet Answer (5 Seconds Of Silence After Detecting Ringbacks)
		! = Flash (1/2 Second On-Hook)
E0		Command Character Echo Disabled
E1	*	Command Character Echo Enabled
F0		Half-Duplex (On-Line Local Echo)
F1	*	Full-Duplex Mode (Normal)
H0	*	Force Modem On-Hook (Off-Line)
H1		Force Modem Off-Hook (Make Busy)
I0		Output Product Code To DTE
I1		Output Product Checksum To DTE
I2		Product Checksum Calculation (Result Code = OK
I9		Output Product Identifier To DTE
L0 L1 L2 L3		Speaker Volume Commands Not Supported - Result Code = OK
M0 M1 M2 M3		Speaker Control Commands Not Supported - Result Code = OK
O0		Return To On-Line Communication
O1		Return To On-Line Communication And Initiate Retrain
Q0	*	Result Code Messages Sent To DTE
Q1		Result Code Messages Not Sent To DTE

Command Summary (Cont'd)

Command		Description (* denotes the factory setting)
Sr?		Read And Display Contents of Register (r)
Sr=n		Set Register (r) To Value (n)
V0		Numeric Result Code Message Format
V1	*	Verbal Result Code Message Format
X0		Basic Status Set/Blind Dialing
X1		Extended Status Set/Blind Dialing
X2		Extended Status Set/Dialtone Detection
X3		Extended Status Set/Blind Dialing/Busy Tone Detection
X4	*	Extended Status Set/Dialtone & Busy Tone Detection
Y0	*	Disable Long Space Disconnect
Y1		Enable Long Space Disconnect
Z		Reset - Load User Stored Settings
+++	*	Escape Code From On-Line To Command State
&C0	*	Force DCD Active
&C1		DCD Indicates On-Line Data Carrier Detected
&C9		Force DCD Active Except Turn Off for One Second After Hang-Up
&D0	*	Force DTR Active
&D1		Enter Command State When DTR Turns Off
&D2		Enter Command State & Hang Up When DTR Turns Off
&D3		Reset To User Stored Settings When DTR Turns Off
&D8		DTR Controlled Dial Stored Number & Hang Up (Async Mode)
&D9		DTR Controlled Data/Talk Switching (Async Mode)
&F		Load Factory Settings
&G0, &G1, &G2		Guard Tone Commands Not Supported-Result Code = OK
&J0	*	RJ-11 Telephone Interface Jack
&J1		RJ-12/13 Type Jack Not Supported - Result Code = OK
&L0	*	Dial Line Mode
&L1		Leased Line Mode

Command Summary (Cont'd)

Command		Description (* denotes the factory setting)
&M0	*	Async Command & Async Communications Mode
&M1		Async Command & Sync Communications Mode
&M2		DTR Controlled Dial Stored Number & Hang-Up (Sync Mode)
&M3		DTR Controlled Data/Talk Switching (Sync Mode)
&P0	*	39% Make, 61% Break Pulse Dial Ratio (USA)
&P1		33% Make, 67% Break Pulse Dial Ratio (UK)
&R0	*	CTS Tracks RTS (Sync Only)
&R1		Ignore RTS (CTS Turns Off Only During Proto.)
&R9		Force CTS Active & Ignore RTS
&S0	*	Force DSR Active
&S1		DSR Indicates On-Line in Data Mode
&T0	*	Terminate Test In Progress
&T1		Enter Local Analog Loopback Test Mode
&T3		Enter Local Digital Loopback Test Mode
&T4		Respond To Remote Digital Loopback Request Signal
&T5		Ignore Remote Digital Loopback Request Signal
&T6		Initiate Remote Digital Loopback Test Mode
&T7		Initiate Remote Digital Loopback w/Self-Test
&T8		Enter Local Analog Loopback w/Self-Test
&W		Write User Configuration To Non-Volatile Mem.
&X0	*	Internal Synchronous Timing
&X1		External Synchronous Timing
&X2		Slave Synchronous Timing
&Zn		Write Telephone Number (n) To Non-Volatile Memory
\A0		64 Character Maximum MNP Stream Size
\A1		128 Character Maximum MNP Stream Size
\A2		192 Character Maximum MNP Stream Size
\A3	*	256 Character Maximum MNP Stream Size
\J0	*	Switched DTE Baud rate
\J1		Fixed DTE Baud Rate
\K0	*	Normal MNP Break Type
\K1		Expedited MNP Break Type
\K2		Destructive Expedited MNP Break Type

Command Summary (Cont'd)

Command		Description (* denotes the factory setting)
\L0	*	Stream MNP Link Mode
\L1		Block MNP Link Mode
\N0	*	Standard Connections Only
\N1		Standard Connections Only
\N2		Reliable-Only Connections
\N3		Auto-reliable Connections
\O		Initiate reliable Link
\Q0		XON/XOFF (Software) Flow Control
\Q1		XON/XOFF (Software) Flow Control
\Q2	*	CTS Signal (Hardware) Flow Control
\Q3		CTS Signal (Hardware) Flow Control
\S		Status: Configuration & Command Listing
\Tn		Inactivity Timer (n = 0* to 15, 5 Minutes Per Count)
\X0	*	XON/XOFF Processed/Not Passed Through
\X1		XON/XOFF Processed & Passed Through
\X2		XON/XOFF Passed Through/Not Processed
\U		Accept Reliable Link
\Y		Switch To Reliable Link

S Register Summary

Register	Default	Function	Range	Units
S0	000	Number Of Rings To Detect Before Autoanswering	000-255	Rings
S1	000	Ring Counter	000-255	Rings
S2	043	Escape Character Code	000-127	ASCII
S3	013	Carriage Return Character Code	000-127	ASCII
S4	010	Line Feed Character Code	000-127	ASCII
S5	008	Backspace Character Code	000-127	ASCII
S6	002	Wait Time For Dialtone	000-255	Second
S7	060	Wait Time For Carrier	001-060	Second
S8	002	Pause Command Time	000-255	Second
S9	006	Carrier Detect Response Time	001-255	0.1 Second
S10	014	Loss-Of-Carrier Disconnect Time-Delay	001-255	0.1 Second
S11	---	Not Used	--------	--------
S12	050	Escape Code Guard Time	020-255	0.02 Second
S13	---	Not Used	--------	--------
S14	170	Bit Mapped Options Register	------	--------
S15	---	Not Used ---------------	--------	--------
S16	000	Modem Test Options Register	------	--------
S17	---	Not Used	--------	--------
S18	000	Test Timer	000-255	Seconds
S19	---	Not Used	--------	--------
S20	---	Not Used	--------	--------

S Register Summary (cont'd)

Register	Default	Function	Range	Units
S21	000	Bit Mapped Options Register	------	-------
S22	118	Bit Mapped Options Register	------	-------
S23	007	Bit Mapped Options Register	------	-------
S24	---	Not Used	-------	-------
S25	005	Delay To DTR (Synchronous Modes Only)	000-255	Seconds
S26	001	RTS/CTS Time-Delay (Synchronous Modes Only)	001-255	0.01 Seconds
S27	064	Bit Mapped Options Register	------	-------
S36	000	Bit Mapped Options Register	------	-------
S37	192	Bit Mapped Options Register	------	-------
S38	000	Bit Mapped Options Register	------	-------
S39	000	Bit Mapped Options Register	------	-------

Appendix E

Quick Reference

for

Debug Commands

1.0 TU Diagnostics

1. Turn power switch ON, verify the following prompt on the display:

```
FPC passed test
PMMU passed test
130Bug>
```

2. Enter "SD" on the keyboard, hit enter and verify the following prompt on the display:

```
130Diag>
```

3. Perform the following diagnostic tests:

Final System Acceptance Test

FATPKG1 Passed _______

VME Bus and VSB Bus Tests

BUS A Passed _______
BUS B Passed _______
BUS C Passed _______
BUS D Passed _______
BUS E Passed _______
BUS F Passed _______
BUS P Passed _______
BUS Q Passed _______
BUS R Passed _______
BUS S Passed _______
BUS T Passed _______

Processor (On Board) Cache Tests

CA20 F Passed _______

CA20 G Passed _______

CA20 H Passed _______

CA20 I Passed _______

CIO (Counter/Timer) Test

CIO A Passed _______
CIO B Passed _______
CIO C Passed _______
CIO D Passed _______
CIO F Passed _______

Memory Management Unit (MMU) Tests

MMU A Passed ________
MMU B Passed ________
MMU C Passed ________
MMU D Passed ________
MMU E Passed ________
MMU F Passed ________
MMU G Passed ________
MMU H Passed ________
MMU I Passed ________
MMU J Passed ________
MMU K Passed ________
MMU L Passed ________
MMU M Passed ________
MMU P Passed ________
MMU Q Passed ________
MMU R Passed ________
MMU S Passed ________
MMU T Passed ________
MMU U Passed ________
MMU V Passed ________
MMU W Passed ________
MMU X Passed ________
MMU Y Passed ________
MMU Z Passed ________
MMU Z0 Passed ________
MMU Z1 Passed ________
MMU Z2 Passed ________
MMU 0 Passed ________

RAM (Off Board) Tests

MT A ________ Passed ________

MT B ________ Passed ________

MT C ________ Passed ________

MT D ________ Passed ________

MT E Passed ________

MT F Passed ________

MT G Passed ________

MT H Passed ________

MT I Passed ________

MT J Passed ______

Appendix F

VI Editor Command Summary

VI EDITOR COMMAND SUMMARY

Sample commands

h j k l	arrow keys move the cursor
i*text*ESC	insert text *abc*
cw*new*ESC	change word to *new*
easESC	pluralize word
x	delete a character
dw	delete a word
dd	delete a line
3dd	... 3 lines
u	undo previous change
ZZ	exit vi, saving changes
:q!CR	quit, discarding changes
/*text*CR	search for *text*
^U ^D	scroll up or down
:*ex cmd*CR	any ex or ed command

Counts before vi commands

Numbers may be typed as a prefix to some commands. They are interpreted in one of these ways.

line/column number	z G |
scroll amount	^D ^U
repeat effect	most of the rest

Interrupting, canceling

ESC	end insert or incomplete cmd
^?	(delete or rubout) interrupts
^L	reprint screen if ^? scrambles it
^R	reprint screen if ^L is -> key

File manipulation

:wCR	write back changes
:qCR	quit
:q!CR	quit, discard changes
:e *name***CR**	edit file *name*
:e!CR	reedit, discard changes
:e + *name***CR**	edit, starting at end
:e + *n***CR**	edit starting at line *n*
:e #CR	edit alternate file
	synonym for **:e #**
:w *name***CR**	write file *name*
:w! *name***CR**	overwrite file *name*
:shCR	run shell, then return
:!*cmd***CR**	run *cmd*, then return
:nCR	edit next file in arglist
:n *args***CR**	specify new arglist
^G	show current file and line
:ta *tag***CR**	to tag file entry *tag*
^]	**:ta**, following word is *tag*

In general, any *ex* or *ed* command (such as *substitute* or *global*) may be typed, preceded by a colon and followed by a CR.

Positioning within file

^F	forward screen
^B	backward screen
^D	scroll down half screen
^U	scroll up half screen
G	go to specified line (end default)
/*pat*	next line matching *pat*
?*pat*	prev line matching *pat*
n	repeat last / or ?
N	reverse last / or ?
/*pat*/+*n*	nth line after *pat*
?*pat*?-*n*	nth line before *pat*
]]	next section/function
[[	previous section/function
(	beginning of sentence
)	end of sentence
{	beginning of paragraph
}	end of paragraph
%	find matching ( ) { or }

Adjusting the screen

^L	clear and redraw
^R	retype, eliminate @ lines
zCR	redraw, current at window top
z-CR	... at bottom
z.CR	... at center
/*pat*/z-CR	*pat* line at bottom
z*n*.CR	use *n* line window
^E	scroll window down 1 line
^Y	scroll window up 1 line

Marking and returning

\`\`	move cursor to previous context
''	... at first non-white in line
m*x*	mark current position with letter *x*
\`*x*	move cursor to mark *x*
'*x*	... at first non-white in line

Line positioning

H	top line on screen
L	last line on screen
M	middle line on screen
+	next line, at first non-white
-	previous line, at first non-white
CR	return, same as +
j	next line, same column
k	previous line, same column